Universidad Autónoma de Madrid
Escuela politécnica superior

Proyecto fin de carrera

ESTIMACIÓN DE LA DENSIDAD DE PERSONAS BASADO EN SEGMENTACIÓN FRENTE-FONDO Y SEGMENTACIÓN FONDO-PERSONA

Ingeniería de Telecomunicación

Rosely Sánchez Ricardo
Mayo 2015

Estimación de la densidad de personas basado en segmentación frente-fondo y segmentación fondo-persona

AUTOR: Rosely Sánchez Ricardo
TUTOR: Álvaro García Martín
PONENTE: José M. Martínez

Video Processing and Understanding Lab
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Mayo 2015

Resumen

Resumen

En la actualidad, el uso de sistemas de visión artificial ha adquirido una gran importancia debido al avance de las tecnologías de procesamiento digital de imágenes y videos y al abaratamiento de las herramientas de captura. La estimación de densidad de personas como parte de los sistemas de visión artificial cuenta con un importante nicho de mercado en el campo de la video-vigilancia como una herramienta para detectar situaciones anormales en lugares públicos como pueden ser peleas, disturbios, protestas violentas, pánico o congestión. La información de densidad puede ser también útil para definir una estrategia de negocio teniendo en cuenta la distribución de las personas en cualquier lugar público o centro comercial, la distribución de la densidad a lo largo del día, etc.

Se han implementado hasta ahora un gran número de estimadores de densidad y una parte importante de ellos se basa en la extracción del fondo de la imagen y la posterior extracción de características de los píxeles del frente de la imagen. Hasta el momento, todos ellos han empleado segmentadores frente-fondo con resultados bastante satisfactorios en los escenarios estudiados. En este proyecto se introducirá el uso de segmentadores persona-fondo para la estimación de densidad de personas. Para ello, se implementará uno de los algoritmos de estimación de densidad del Estado del Arte y se compararán los resultados obtenidos al emplear un segmentador frente-fondo y un segmentador fondo-persona en diversos escenarios.

Palabras Clave

Estimación de densidad de personas, segmentadores frente-fondo, segmentadores persona-fondo, contar personas en una imagen.

Abstract

Currently, the use of artificial vision systems has acquired great relevance due to the advancement of digital image and video processing technologies and the cheapening of capture tools. The crowd density estimation as part of artificial vision systems has an important niche market in video-surveillance. The crowd density estimation is an important tool to detect abnormal situations in public places such as fights, disturbances, violent protests, panic or congestion. Density information could be also helpful for creating a business strategy according to the distribution of people in plublic places or shopping centers and the distribution of people over time.

So far, a large number of crowd density estimators has been implemented. A significant part of these estimators use background substration and extract features from foreground pixels. All of them use foreground-background segmentation getting good results for the studied scenarios. This proyect will introduce the use of people-background segmentation for crowd density estimation. With the goal of comparing both types of segmentation, one algorithm of the Estate of the Art for density estimation will be implemented and then, results for both types of sementation, foreground-background and people-background segmentation, will be compared in several scenarios.

Key words

Crowd density estimation, foreground-background segmentation, people-background segmentation, count people in images.

Agradecimientos

Con motivo de la lectura de mi Proyecto de Fin de Carrera, y por tanto del final de mis estudios, me gustaría agradecer a todas aquellas personas que de alguna manera hayan contribuido a que esté hoy aquí.

Comenzando con las personas que han participado de manera directa en la realización de este proyecto, quiero en primer lugar dar las gracias a MIGUEL, por ser quien más me ha animado y motivado tanto a sacar adelante este PFC, como la carrera en general. Gracias por tu ayuda, por ser una de las personas mejor conoce este proyecto, por facilitarme tu ordenador, que ha minimizado al máximo las largas horas de ejecución, pero sobretodo, gracias por estar a mi lado durante prácticamente toda la carrera. Tq.

En segundo lugar, me gustaría agradecer a mi TUTOR, Álvaro Grarcía-Martín, que sin duda ha puesto todos los medios necesarios y ha hecho su labor mejor que bien. Gracias por hacer las cosas tan fácil siendo tan claro y directo, tan accesible, por la inmediatez de tus respuestas y correcciones y por tu ayuda, no sólo en lo meramente académico, sino también con todas las diligencias burocráticas necesarias a lo largo de todo el PFC.

Quiero agradecer a todos los que, aunque no de una manera tan direccta, también han sido parte de este proyecto. A mi ponente, J. M. Martínez y a todos los miembros del VPULab. Creo que yo y varias generaciones de estudiantes os agredecemos que os toméis tan en serio los PFCs. Gracias por vuestra ayuda tanto a la hora de elegir el proyecto como en el día a día, poniendo a nuestra disposición los medios del laboratorio para la realización del mismo.

Por último, gracias a todas las personas con las que he compartido a lo largo de estos años, mis compañeros, porque o bien me habéis ayudado con las prácticas y las asignaturas, o bien habéis estado conmigo compartiendo los mejores recuerdos que guardaré de mis

años de estudiante.

Finalmente, y no por ser menos importante, quiero agradecer a los principales responsables de que esté hoy aquí, a mi FAMILIA y en especial a mis PADRES, por poner siempre todo y más de lo que ha estado a su alcance para que yo tenga la mejor vida posible. Gracias por dar a mis estudios la prioridad que merecían y por darme vuestro apoyo a todos los niveles para conseguirlo, tanto por el sacrificio personal como económico que ello os ha supuesto. Gracias, os quiero.

Índice general

Índice de figuras

Índice de cuadros

1 Introducción

1.1. Motivación

En la actualidad, el uso de sistemas de visión artificial ha adquirido una gran importancia en múltiples ámbitos. Este crecimiento se debe principalmente a dos factores: el avance que ha experimentado el procesamiento digital de imágenes y vídeo en los últimos años, y el abaratamiento de precios de las herramientas de captura de imágenes y vídeo.

La gran implantación de las cámaras de vídeo en la sociedad en la que vivimos hace que sea inviable tener personal suficiente para controlar y gestionar las grabaciones realizadas. Por este motivo, las técnicas de monitorización automática de escenas en las que hay personas, tales como la detección automática de objetos, el seguimiento, el reconocimiento de acciones y demás tecnologías aplicadas al análisis y comprensión de la imagen digital, han adquirido un gran peso e importancia.

La utilización de estos sistemas de monitorización automáticos tiene un importante nicho de mercado en el mundo de la seguridad, ya que se utilizan en sistemas de vídeo-vigilancia para identificar cualquier tipo de situación anormal en lugares públicos, como pueden ser estadios, aeropuertos, plazas, terminales, etc. Las situaciones más habituales suelen ser peleas, disturbios, protestas violentas, pánico, congestión o cualquier otra anomalía. Este

proyecto se centrará en la estimación de densidad de personas, que constituye una de las tareas primordiales de la monitorización de multitudes, ya que es el indicador más común de tales comportamientos.

El análisis de multitudes utilizando la visión artificial, es también muy importante a la hora de decidir una estrategia de negocio. Por ejemplo, la distribución de las personas en un centro comercial puede servir para identificar las preferencias de los clientes, mientras que la cantidad total de personas permitiría analizar el funcionamiento general del centro.

Existen actualmente numerosos métodos y algoritmos distintos para estimar la densidad de personas en vídeos. Muchos de ellos buscan solventar problemas como la oclusión entre personas cercanas, cambios de iluminación o los efectos de la perspectiva y la orientación de la cámara en la estimación. Hay en la literatura numerosos enfoques y métodos, aunque una parte importante de los mismos se basan en la extracción de características locales sobre la máscara resultante de la segmentación de la imagen. El objetivo de este proyecto es implementar un estimador de densidad de personas basado en el algoritmo propuesto en [83] y comparar los resultados obtenidos de la estimación utilizando dos tipos de segmentadores:

1. Segmentador frente-fondo: Este proyecto se ha centrado en los segmentadores de extracción o modelado de fondo (**BGS**). El objetivo principal de estos segmentadores es separar el primer plano de la imagen del fondo, para lo cual, se genera un modelo de fondo que permite separar los objetos en movimiento del resto. Hay en la literatura un gran número de algoritmos de modelado de fondo [7], de los cuales, se deberá elegir uno para la estimación de densidad.

2. Segmentador fondo-persona: El objetivo de los segmentadores persona-fondo es determinar las zonas de la imagen que no contienen personas y que por tanto se clasifican como fondo. Hay un único segmentador de este tipo en la literatura [31].

1.2. Objetivos

El objetivo de este proyecto es implementar un algoritmo de estimación de densidad de personas y comparar los resultados obtenidos en la estimación al utilizar un algoritmo de segmentación frente-fondo y un algoritmo de segmentación persona-fondo.

Para ello, previamente se realizará un estudio del Estado del Arte de estimación de densidad de personas y de ambos tipos de segmentación. A continuación, se elegirá un algoritmo de estimación basado en segmentación y se implementará utilizando dos algoritmos de segmentación: uno de segmentación frente-fondo y otro de segmentación persona-fondo. Por último, se evaluarán y presentarán los resultados obtenidos con ambos segmentadores de manera comparativa. Para la evaluación se deberán definir previamente las bases de datos a utilizar y la métrica para medir los resultados.

De manera más detallada, se presentan a continuación cada uno de los objetivos antes mencionados:

1. Estudio del Estado del Arte de estimación de densidad de personas.
2. Elección de algoritmo de estimación de densidad basado en segmentación.
3. Estudio del Estado del Arte de segmentación frente-fondo.
4. Elección de algoritmo de segmentación frente-fondo.
5. Estudio del Estado del Arte de segmentación persona-fondo.
6. Elección de algoritmo de segmentación persona-fondo.
7. Implementación de algoritmo de estimación de densidad de personas.
8. Selección de las bases de datos para la evaluación del algoritmo implementado con ambos segmentadores.
9. Definición de métrica para la evaluación de los resultados.
10. Exposición y comparación de los resultados empleando segmentación frente-fondo y segmentación persona-fondo.
11. Elaborar conclusiones del proyecto y propuestas de trabajo futuro.
12. Elaborar la memoria del proyecto.

1.3. Medios

Los medios necesarios para la realización de este proyecto han sido facilitados por el grupo de investigación Video Processing and UnderstadingLab (VPULab) del Departamento

de Tecnología Electrónica y de las Comunicaciones de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid. Los principales elementos utilizados para la realización de este proyecto han sido:

1. Parque de PC's (Windows/Linux) interconectados a través de la red de área local y con acceso a Internet y a los servidores del VPULab.

2. Software para el desarrollo del proyecto, Matlab y Visual Studio.

3. Bases de datos de secuencias de vídeo.

1.4. Estructura de la memoria

La memoria del proyecto tendrá la siguiente estructura:

- Introducción.
- Estado del arte.
- Algoritmo.
- Evaluación.
- Conclusiones y trabajo futuro.

2

Estado del arte

2.1. Introducción

En este capítulo se estudiará el Estado del Arte relacionado con este trabajo. Para ello, en la sección 2.2 se presenta un resumen de los distintos métodos y enfoques que se han utilizado para estimar la densidad de personas.

Dado que el algoritmo de estimación de densidad de personas implementado en este proyecto, parte de la máscara resultante de la extracción del fondo de la imagen, también ha sido necesario realizar un estudio de los distintos métodos de segmentación. Dicho estudio se puede ver en la sección 2.3.

2.2. Estimación de densidad de personas

En esta sección se presenta un estudio del Estado del Arte sobre los estimadores de densidad de personas.

Existen numerosos métodos en la literatura para estimar la cantidad de personas presentes en una imagen que emplean técnicas de visión artificial. La mayor parte de ellos

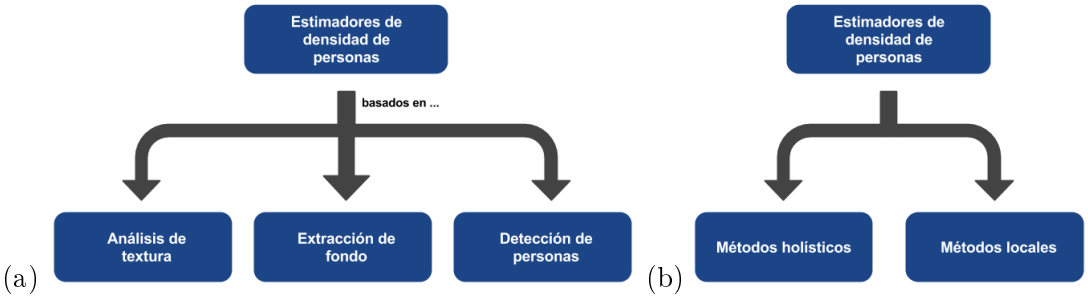

Figura 2.1: Clasificación de estimadores de densidad de personas. (a) y (b) representan dos clasificaciones distintas.

extraen diferentes características de la imagen y a continuación emplean una técnica de aprendizaje supervisado para inferir la densidad a partir de dichas características.

Sin embargo, se pueden diferenciar claramente tres tipos de estimadores (ver figura 2.1 (a)): el primero, utiliza características de textura de la imagen, por lo que se les puede denominar métodos basados en análisis de textura; el segundo, emplea diversas características extraídas de los píxeles del frente de la imagen (*foreground*) y se les conoce como métodos basados en extracción de fondo y el último grupo, incluye a los estimadores que utilizan detectores o características que son específicas o propias de individuos o grupos, a los que se les llamará en este proyecto, métodos basados en detección de personas. Cada uno de ellos se describirá en las secciones 2.2.1, 2.2.2 y 2.2.3 respectivamente.

A su vez, existe otra clasificación, no menos importante que diferencia solamente dos tipos de estimadores (ver figura 2.1 (b)): estimadores de tipo holístico que emplean características holísticas de la imagen y estimadores locales, que emplean características locales. En la sección 2.2.4 se presentará un resumen de las ventajas y desventajas de cada enfoque y se clasificarán los estimadores mencionados en las secciones anteriores (2.2.1, 2.2.2 y 2.2.3).

2.2.1. Métodos basados en análisis de textura

El principio fundamental de los métodos basados en análisis de textura [64] es que las regiones de una imagen con alta densidad presentan patrones finos de textura, mientras que las regiones de una imagen con baja densidad presentan patrones gruesos de textura. Por este motivo, se emplean descriptores de textura para medir el grosor de dichos

patrones y a partir de ahí estimar la densidad de la imagen.

Hay numerosos ejemplos en la literatura de estimadores de densidad que utilizan análisis de textura como [64, 65, 66, 77, 95]. Para ello, en primer lugar se extraen las características de textura y a continuación se emplea un clasificador para determinar el nivel de densidad. Los niveles de densidad son habitualmente los siguientes: densidad muy baja (0 - 15 personas), densidad baja (16 - 30 personas), densidad moderada (31 - 45 personas), densidad alta (46 - 60 personas) y densidad muy alta (más de 60 personas).

Los métodos de análisis de textura se pueden clasificar en métodos estadísticos, métodos estructurales, métodos espectrales y métodos basados en un modelo [69]. Los principales ejemplos de métodos estadísticos que encontramos en la literatura se basan, o bien, en la elaboración de una matriz de dependencia de niveles de gris, también llamada matriz de co-ocurrencia [64, 66, 77, 95] o bien, en los momentos invariantes ortogonales de Chebyshev [77]. Como ejemplo de método estructural, Marana en [64] utiliza estadísticas de primer orden extraídas tras aplicar la transformada de Hough y lo denomina segmentos de línea recta. También Marana en [64] utiliza el espectro de Fourier como método espectral basado en la idea de que la energía espectral de patrones gruesos de textura se concentra en las bajas frecuencias mientras que la energía de patrones finos de textura se concentra en las altas frecuencias. Por último, el método basado en modelo más ampliamente utilizado en el Estado del Arte se conoce como dimensión fractal de Minkowski [64, 65, 77] y permite clasificar las imágenes por su nivel de densidad a partir de su dimensión fractal D, estimada con el método propuesto por Minkowski.

Existen varios artículos en el Estado del Arte que realizan una comparativa de distintos métodos de análisis de textura así como de los distintos clasificadores. Atendiendo solamente a los métodos de análisis de textura, Marana en [64] realiza una comparativa de cuatro de los métodos antes mencionados, concretamente uno de cada tipo: matriz de dependencia de niveles de gris, segmentos de línea recta, espectro de Fourier y dimensión fractal de Minkowski. De dicha comparativa, el autor concluye que el método de matriz de co-ocurrencia funciona ligeramente mejor que los otros tres, siendo el funcionamiento del resto muy similar. También H. Rahmalan en [77] compara tres métodos de análisis te textura: matriz de dependencia de niveles de gris, dimensión fractal de Minkowski y momentos invariantes ortogonales de Chebyshev. Esta vez, los resultados muestran que el método que peor funciona es el de dimensión fractal de Minkowski y el mejor los mo-

mentos invariantes ortogonales de Chebyshev, aunque seguido muy de cerca por la matriz de co-ocurrencia. Por tanto, ambas comparativas muestran que de los dos métodos más utilizados: matriz de co-ocurrencia y dimensión fractal de Minkowski, el primero ofrece una estimación más exacta que el segundo.

Atendiendo a los clasificadores empleados para determinar el nivel de densidad a partir de las características de textura, los más utilizados son: mapa de auto-organizado (*Self Organizing Map*) [64, 65, 66, 77], funciones de ajuste (*fitting functions*) [64, 65], clasificador Bayesiano [64, 65] y máquinas de soporte vectorial [95] (*Support Vector Machines*). En [64] y [65] Marana compara el funcionamiento de los distintos clasificadores y concluye que el clasificador empleado no es crucial ya que en media producen resultados similares.

De manera general, los métodos basados en análisis de textura funcionan bien para clasificar las imágenes de acuerdo a su nivel de densidad con porcentajes de acierto alrededor del 70 % - 80 % para cinco niveles de densidad. Sin embargo, presentan serios problemas al funcionar en exteriores debido a cambios de iluminación, a la presencia de sombras o a la textura de la superficie del suelo. En este sentido, H. Rahmalan en [77] estudia los efectos de los cambios de iluminación en exteriores extrayendo resultados con imágenes tomadas durante la mañana (grandes cambios de iluminación), durante la tarde (cambios de iluminación menores) y una combinación de ambas. Los resultados revelan que efectivamente los estimadores estudiados funcionan mejor con las imágenes tomadas durante la tarde, ya que presentan menores cambios de iluminación. Además, al igual que al resto de estimadores de densidad, a los basados en análisis de textura también les afecta la oclusión de las personas que van en grupos.

2.2.2. Métodos basados en extracción de fondo

Los estimadores de densidad basados en extracción de fondo se caracterizan principalmente por tener una fase previa a la extracción de características en la que se realiza un procesado de la imagen que permite la extracción del fondo o *background* (segmentación). A continuación, se realiza un resumen de todas las etapas que tienen en común estos estimadores [101]:

- **Extracción del fondo de la imagen**

  En esta etapa, se obtiene una imagen binaria con los píxeles del frente de la ima-

gen original (*foreground*) blancos y los píxeles del fondo de la imagen original (*background*) negros. Esta imagen binaria se divide en segmentos (*blobs*) que están constituidos por un conjunto de píxeles conectados y que representan a los individuos o grupos de personas. El segmentador que se emplea para modelar el fondo de la imagen puede ser de dos tipos: segmentador frente - fondo o segmentador persona - fondo (ver Estado del Arte de segmentación en sección 2.3). Se puede decir que el segundo tipo es, a priori, más apropiado para determinar la densidad ya que es teóricamente más sencillo inferir el número de personas de la escena a partir de los píxeles clasificados como persona que a partir de los píxeles del frente de la imagen (no necesariamente personas). No obstante, todos los estimadores de densidad implementados hasta el momento utilizan segmentadores frente - fondo y han demostrado que en un gran número de escenarios el resultado es más que aceptable. Precisamente, en este proyecto se intentará comparar la utilización de ambos tipos de segmentación mediante la evaluación de los resultados de estimación de densidad en escenarios muy diversos.

- **Extracción de características**

  En esta etapa se extraen las características de los píxeles del frente de la imagen. Se suele combinar el uso de varias características distintas para dotar al estimador de mayor robustez. Las características más utilizadas son el área [11, 16, 19, 38, 47, 60, 74, 82, 83], el número de píxeles de borde [16, 19, 82], el histograma de orientación de bordes [11, 47, 82, 83] y el perímetro [11, 82, 83]. Chan en [11] utiliza también características de textura (dimensión fractal de Minkowski y matriz de co-ocurrencia de niveles de gris). Sin embargo, su algoritmo se ha incluido en este grupo debido a que las características se calculan sobre los píxeles del frente de la imagen, que a su vez, se han dividido en dos máscaras diferentes según la dirección del movimiento de los peatones. Tal y como se explicará en la sección 2.2.4, estas características se pueden estimar sobre el conjunto de los píxeles del frente de la imagen (características holísticas) o sobre cada segmento (*blob*) por separado (características locales).

- **Normalización de características**

  Se ha demostrado que la relación entre el número de personas dentro de la escena y el número total de píxeles del frente de la imagen es aproximadamente lineal [74],

por lo que esta etapa aparece solo en algunos ejemplos del Estado del Arte. Sin embargo, la extracción de fondo y la regresión lineal no es suficiente para estimar la densidad debido a los efectos de la perspectiva y la oclusión [83].

El efecto de la perspectiva hace que los objetos cercanos a la cámara aparezcan más grandes y por tanto cualquier característica extraída de los píxeles del frente de la imagen representará una porción más pequeña de dicho objeto que una extraída de un objeto más lejano. Además, el ángulo de observación de la cámara a un objeto con respecto a el plano de tierra no es constante dentro de la imagen y suele variar mucho cuando la cámara se sitúa cerca de la escena. Esto hace que sea necesario normalizar las características para corregir el efecto de la perspectiva. Para ello, la solución más habitual suele ser elaborar un mapa de normalización de perspectiva para asignar un peso $W(i,j)$ a cada píxel de la imagen atendiendo a su distancia a la cámara. Finalmente, a las características de dos dimensiones se les aplica directamente el peso $W(i,j)$, mientras que a las características de una dimensión se les aplica la raíz cuadrada del peso $\sqrt{W(i,j)}$ [11, 38, 60, 82].

Hay otro tipo de distorsión geométrica que se debe al cambio de configuración de la cámara. El objetivo de añadir este factor también a la normalización de características es tener características que sean, no solo invariantes a traslaciones de los peatones, sino también al punto de vista de la cámara. No tener en cuenta este factor implicaría tener que realizar el entrenamiento cada vez que cambiase el punto de vista. Para corregirlo, algunos autores añaden un nuevo factor a la corrección de perspectiva [47] y otros emplean los parámetros de Tsai de calibración de la cámara para calcular un mapa de densidades que permita corregir tanto la perspectiva como la distorsión introducida por la cámara [83].

- **Regresión**

  En esta etapa se estima la densidad de personas mediante regresión. La regresión permite estimar el número de personas en la escena y no solamente clasificar la imagen por su nivel de densidad, como ocurría con los métodos basados en análisis de textura. Los algoritmos de regresión requieren de una primera fase de entrenamiento que permite determinar la relación entre las características y el número de personas. Una vez conocida la relación, es posible determinar la densidad a partir de las características extraídas. Hay en la literatura numerosos algoritmos de regre-

sión aunque los más utilizados en este caso son la regresión lineal [19, 60, 74], redes neuronales [38, 47, 82] y proceso de Gauss (*Gaussian Process Regression*) [11, 83].

Los algoritmos basados en extracción de fondo funcionan bastante bien incluso en escenarios con elevada densidad, Además, se eliminan las fuentes de error debidas a la textura del fondo de la imagen gracias al proceso de segmentación. Por otro lado, la normalización de las características permite estimar de manera más precisa la relación entre éstas y el número de personas, lo cual a su vez, facilita el entrenamiento.

2.2.3. Métodos basados en detección de personas

En este grupo de estimadores se incluyen todos aquellos que realizan detección o seguimiento de personas utilizando características específicas de individuos o de grupos. Estos algoritmos no tienen, como en los casos anteriores, unas etapas comunes tan definidas, por lo que a continuación se presentarán algunos de los ejemplos más significativos encontrados en el Estado del Arte [40, 83].

Lin en [54] propuso un algoritmo para estimar la densidad de personas en tres fases. En la primera fase, se buscan objetos con contornos de cabezas en la imagen mediante la trasformada wavelet de Haar. En la segunda, las características del objeto se analizan mediante una máquina de soporte vectorial (*Support Vector Machine*) con el objetivo de clasificarlo como cabeza o no. Finalmente, se realiza una transformación de perspectiva que permite estimar mejor la densidad de la imagen. La principal ventaja de este algoritmo es que puede ser empleado en escenas de alta densidad cuando solo las cabezas de las personas son visibles.

Zhao y Nevatia en [105] propusieron un algoritmo para segmentar personas. En su trabajo, se utilizan varios modelos de personas 3D para representar los píxeles del frente de la imagen y un modelo probabilístico basado en cadenas de Markov Monte Carlo permite integrar en un sistema Bayesiano características como la forma del cuerpo, la altura, el modelo de la cámara, posibles cabezas, etc. Sin embargo, en un entorno de alta densidad la representación de todo el cuerpo no suele ser muy útil debido a grandes oclusiones.

Leibe [49] propuso un esquema de detección de peatones basado en un segmentador. Éste, combina características locales (un modelo de forma invariante en escala) y características globales (distancia de Chamfer) para calcular la probabilidad de que haya una

persona. El algoritmo propuesto es capaz de detectar personas en escenas con densidades relativamente elevadas y grandes oclusiones.

Lempitsky [50] propuso un algoritmo para contar objetos que busca una función de densidad F, como función de la intensidad de los píxeles de la imagen, de manera que, integrar dicha función sobre cualquier región, dé como resultado el número de objetos de dicha región. Se trata de un enfoque local en el que cada píxel p se representa por un vector de características $x_p$que contiene información del frente de la imagen e información de gradiante. Se utiliza un modelo lineal para obtener la densidad en cada píxel, $F(p) = W^T x_p$, por lo que el número de objetos se obtiene integrando F sobre una región de interés R: $\sum_{p \epsilon R} F(p)$.

Rabaud y Belongie [76] desarrollaron un algoritmo para segmentar individuos en una multitud. Ellos emplearon el algoritmo de seguimiento de características Kanade-Lucas-Tomasi para detectar objetos que se mueven en la escena. El algoritmo de seguimiento se combina con un filtro temporal y espacial y se utiliza un algoritmo de agrupamiento (en inglés, *clustering*) para agrupar características similares en una trayectoria relativa a un único objeto. Los autores validaron los resultados empleando 3 bases de datos distintas e incluso con una secuencia de vídeo de células para demostrar la robustez del algoritmo segmentando cualquier tipo de individuos. Una limitación clara de este método es la detección de multitudes estacionarias donde no hay movimiento.

Brostow y Cipolla [8] presentaron un algoritmo de agrupamiento de aprendizaje no supervisado Bayesiano para detectar movimientos independientes en multitudes. Su hipótesis radica en que un par de puntos que se mueven juntos deberían ser parte de una misma entidad. Un algoritmo de flujo óptico combinado con una búsqueda exhaustiva basada en la correlación cruzada normalizada se emplea para el seguimiento de algunas características de la imagen. Posteriormente, se aplica el algoritmo de aprendizaje no supervisado para agrupar dichas características con el objetivo de identificar movimientos individuales. Uno de los puntos más interesantes de este método es que no requiere entrenamiento o modelo de apariencia para seguir o identificar a los individuos.

Jones y Snow [42] desarrollaron un clasificador para detectar peatones usando información espacio temporal. Para ello usan tres tipos de características: características *Haar-like* aplicadas directamente sobre cada *frame*, diferencia absoluta de características *Haar-like*

en *frames* adyacentes y un filtro de diferencia desplazado para capturar el movimiento de los peatones (se emplearon 8 direcciones de desplazamiento). A continuación, se utiliza el algoritmo *Adaboost* (*Adaptive Boosting*) para construir un clasificador *Soft Cascade* basado en un set de imágenes de entrenamiento etiquetadas manualmente (en total se entrenaron 8 clasificadores para las 8 direcciones de movimiento). En general, este algoritmo diferencia bastante bien personas de vehículos pero tiende a fallar en escenarios de alta densidad. Además, los cambios en la configuración de la cámara requieren un nuevo entrenamiento ya que la apariencia de las personas depende de la posición de la misma.

En términos generales, tal y como se ha visto, estos algoritmos de detección, aplicados a la estimación de densidad de personas tienen como principal ventaja el hecho de son capaces no sólo de aportar información de cantidad, sino también de la posición y distribución de las personas dentro de la imagen. Sin embargo, suelen ser poco apropiados para lidiar con entornos de alta densidad en los que la oclusión entre personas dificulta su detección.

2.2.4. Métodos holísticos vs métodos locales

Se consideran métodos holísticos a aquellos que emplean características holísticas, para estimar la densidad y métodos locales a aquellos que utilizan características locales o referidas a un solo grupo o individuo. Todos los estimadores basados en análisis de textura y todos los estimadores basados en extracción de fondo a excepción de [82, 83] son de tipo holístico, mientras que todos los estimadores basados en detección de personas y los dos antes mencionados [82, 83] son de tipo local.

Aunque la densidad de personas de una imagen constituye una característica holística de la misma, los métodos que emplean características holísticas para estimar la densidad requieren normalmente de una mayor cantidad de datos de entrenamiento para aprender todas las posibles distribuciones, comportamientos y densidades dentro de la escena, lo cual además, hace necesaria la anotación de un gran número de *frames* [82].

Por este motivo, Ryan en [82, 83] sugiere que la relación entre las características de una imagen y el número de personas es más fiable y consistente a nivel local. Por ejemplo, en una imagen con 20 personas, éstas podrían estar distribuidas en 2 grupos de 10 o en 10 grupos de 2, lo cual desde un punto de vista holístico haría que las características extraídas en ambos casos fueran muy distintas.

2.3. Segmentación

Considerando que uno de los objetivos principales de este proyecto es comparar los resultados obtenidos de la estimación al realizar segmentación frente-fondo y segmentación persona-fondo, en la sección 2.3.1 se describirán los métodos existentes para la segmentación frente-fondo y en la siguiente sección (Segmentación persona-fondo), se presenta el único algoritmo del Estado del Arte que realiza segmentación persona-fondo.

2.3.1. Segmentación frente-fondo

En esta sección, se realiza un estudio del estado del Estado del Arte de segmentación frente-fondo que se centrará específicamente en la extracción de fondo (*Background Substraction* o **BGS**), también llamada detección del frente de la imagen (*Foreground Detection*). Para ello, se ha tomado como fuente principal, el estudio publicado por Bouwmans [7] sobre este tipo de segmentadores que presenta un resumen exhaustivo de todos los algoritmos de modelado de fondo y los clasifica atendiendo al modelo matemático que emplean. A continuación, se presenta un resumen de las características generales de dichos segmentadores (sección 2.3.1.1) y la clasificación de los algoritmos encontrados en el Estado del Arte (sección 2.3.1.2).

2.3.1.1. Características generales

Los algoritmos de extracción o modelado de fondo (**BGS**), tienen como objetivo principal separar el primer plano de la imagen (*foreground*) del fondo (*background*), para lo cual, se genera un modelo de fondo que permite separar los objetos en movimiento del resto.

A continuación, se presenta un resumen de las fases principales que debe tener un algoritmo **BGS:**

1. **Inicialización del modelo**
   Generalmente la inicialización se realiza empleando el primer *frame* o un modelo de fondo aprendido de un conjunto de *frames* de entrenamiento, aunque en escenarios reales no se puede garantizar que estos *frames* no contengan objetos del primer plano de la imagen.

2. **Actualización del modelo**

   El modelo no debe ser estático sino que se debe adaptar a los cambios que van sucediendo en la escena con el paso del tiempo. Hay en la literatura 3 esquemas de actualización: ciego (*blind*), selectivo y adaptativos difusos [22]. El primer esquema de actualización, el ciego, actualiza todos los píxeles, sean de frente o de fondo con una misma regla, que suele ser:

   $$B_{t+1}(x,y) = (1-\alpha)B_t(x,y) + \alpha I_t(x,y) \tag{2.1}$$

   donde α es la tasa de aprendizaje (*learning rate*) que es una constante en $[0,1]$ y B_t e I_t son el fondo y la imagen actual respectivamente. El problema de este esquema es que los píxeles del primer plano se emplean para calcular el nuevo fondo y por tanto lo contaminan. Por este motivo, el esquema selectivo consiste en adaptar el nuevo fondo con diferentes tasas de aprendizaje dependiendo de que el píxel haya sido previamente clasificado como fondo o como frente. La idea es actualizar los píxeles de fondo muy rápidamente y los del frente muy lentamente, por lo que la tasa de aprendizaje para los píxeles del frente de la imagen,β, debe ser mucho menor que la tasa de aprendizaje de los píxeles del fondo, α ($\beta << \alpha$). La principal desventaja de este nuevo esquema es que los errores en la clasificación inicial de los píxeles como frente o fondo, se traducirán en un error permanente. Los esquemas adaptativos difusos abordan este problema mediante la variación de la regla de adaptación en función del resultado de la detección de fondo.

   La tasa de aprendizaje del modelo es un parámetro fundamental que determina la capacidad de adaptación a los cambios de la escena y se puede fijar o ajustar de manera dinámica mediante métodos estáticos o difusos (*fuzzy*).

3. **Detección de frente**

   En esta etapa se clasifican finalmente los píxeles como fondo o frente.

En la figura (2.2) se ilustra el proceso completo, tal y como se ha descrito hasta ahora.

Uno de los aspectos más importantes en el proceso de sustracción de fondo es la elección de las características utilizadas. Las características se pueden clasificar según [52] en características espectrales (color), características espaciales (bordes, textura) y características temporales (movimiento). Todas ellas tienen propiedades diferentes que permiten abordar situaciones críticas distintas como cambios de iluminación, movimiento, cambios

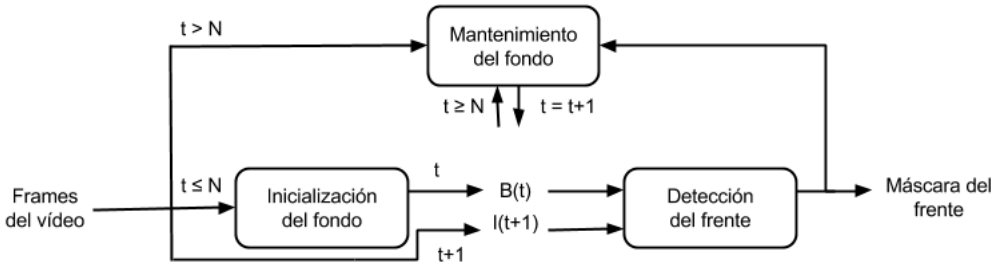

Figura 2.2: Proceso de sustracción de fondo. N es el número de *frames* empleado en la inicialización del modelo. B_t y $I_t$son el fondo y la imagen actual en el momento t, respectivamente. Fuente: [7].

en la estructura del fondo, etc. La utilización de varias características, por tanto, proporciona mayor robustez para afrontar cualquiera de estos cambios.

El desarrollo de un algoritmo de extracción de fondo, requiere del diseño de cada una de las etapas antes mencionadas y de la elección de las características de acuerdo con el tipo de situaciones que se desean solventar.

A continuación se resumen los problemas más frecuentes que aparecen en la extracción de fondo:

- **Imágenes ruidosas:** esto se puede deber a la mala calidad de los dispositivos de captura.
- ***Jitter* de la cámara:** movimiento de la cámara debido al viento o cualquier otro factor.
- **Auto configuración de la cámara:** causa la modificación de la dinámica de los niveles de color entre distintos *frames* de la misma secuencia.
- **Cambios de iluminación:** los cambios de iluminación provocan falsas detecciones en la máscara del frente de la imagen.
- ***Bootstrapping*:** el fondo de la imagen no se puede proporcionar hasta que no finalice el período de entrenamiento.
- **Camuflaje:** las características de píxeles de objetos del frente de la imagen pueden ser incluidas en el modelo de fondo por lo que la clasificación es incorrecta.
- **Apertura del frente de la imagen:** en los objetos con regiones coloreadas unifor-

memente puede no detectarse el movimiento, por lo que se producen falsos negativos en el frente de la imagen.

- **Objetos del fondo movidos:** cuando se mueven objetos del fondo de la imagen se pueden detectar como parte del frente.
- **Nuevos objetos en el fondo de la imagen:** el resultado es el mismo que en el caso anterior.
- **Fondos dinámicos:** los fondos variables como árboles o el agua pueden causar la aparición de falsos positivos.
- **Fantasmas:** cuando un objeto que pertenecía inicialmente al fondo de la imagen se mueve, tanto éste como su posición antigua en el fondo de la imagen, denominada fantasma, son detectados.
- **Frente estático:** si algún objeto del primer plano de la imagen no se mueve, no se puede distinguir del fondo y por tanto se clasifica como tal.
- **Sombras:** las sombras se pueden detectar erróneamente como frente de la imagen.

2.3.1.2. Clasificación

Bouwmans propone una clasificación de los algoritmos **BGS** basada en el modelo matemático que estos emplean [7]. En la tabla 2.1 se muestra una visión general de esta clasificación.

En el apéndice A se presenta un resumen de las principales características de cada uno de los tipos de segmentadores presentados en la tabla 2.1.

De manera general, todos los algoritmos del Estado del Arte pretenden solventar la segmentación mediante modelado de fondo. Sin embargo, prácticamente ninguno de ellos es capaz de afrontar con éxito toda la problemática expuesta en la sección 2.3.1.1 que aparece en escenarios reales. Según el tipo de escena y las condiciones en que se haya capturado la secuencia podrían aparecer diversos factores tales como: ruido, cambios de iluminación, sombras, fondos dinámicos, etc., que afectan de manera significativa los resultados de la segmentación. Por este motivo, estos métodos se encuentran en continua evolución y abordan la problemática del modelado de fondo mediante la aplicación de

Tipo de modelo	**Categorías**	**Subcategorías**
Tradicionales	Modelos Básicos	Media, Mediana, Histograma
		Clasificación por intensidad de píxel
	Modelos Estadísticos	Modelos Gaussianos
		Modelos basados en **SVM**
		Modelos basados en aprendizaje subespacial
	Modelos de Agrupamiento	Modelos *K-means, Codebooks,* agrupamiento secuencial básico
	Modelos de Redes Neuronales	Regresión general (**GNN**), multievaluadas (**MNN**), competitivas (**CNN**), dipolar competitivas (**DCNN**)
		Mapas de autoorganizado (**SONN**), Mapas de autoorganizado de crecimiento jerárquico (**GHSONN**)
	Modelos de Estimación	Filtro de Wiener, filtro de Kalman, filtro de Chebychev
Recientes	Modelos de Fondo Estadísticos Avanzados	Modelos de mezclas
		Modelos híbridos
		Modelos no paramétricos: **ViBe** y **PBAS**
		Modelos multi-kernel
	Modelos Difusos	Modelado de fondo difuso, detección de fondo difusa, actualización del fondo difusa, características difusas, post-procesado difuso
	Modelos de Aprendizaje Subespacial	Discriminatorios, Mixtos
	Modelos Subespaciales Robustos (**RPCA**)	**RPCA** via *Principal Componens Pursuit*, **RPCA** via *Outlier Pursuit*, **RPCA** via *Sparsity Control*
	Modelos de Minimización low-rank (**LRM**)	
	Modelos *Sparse*	*Compressive sensing models*
		Structured sparsity models
		Dynamic group sparsity models
		Dictionary models
	Modelos de Dominio Transformado	Transformada rápida de Fourier (**FFT**)
		Transformada discreta del coseno (**DCT**)
		Transformada de Walsh (**WT**)
		Transformada Wavelet (**WT**)
		Transformada de Hadamard (**HT**)

Cuadro 2.1: Clasificación de los algoritmos de modelado de fondo. Fuente: [7].

diversos y complejos modelos matemáticos que buscan, no sólo una aproximación más precisa, sino también más eficiente y con un coste computacional menor.

2.3.2. Segmentación persona-fondo

El objetivo de los segmentadores persona-fondo es determinar las zonas de la imagen que no contienen personas y que por tanto se clasifican como fondo (*background*). Además, se penalizan mucho más los píxeles de la imagen incorrectamente clasificados como fondo, de manera que la máscara resultante tendrá un sesgo en el fondo de la imagen y no en las personas, como ocurre en los algoritmos basados en detección de personas. Con ello, se pretende garantizar que ningún píxel perteneciente a una persona sea clasificado como fondo, aunque no se garantice lo contrario, es decir, que puede haber píxeles del fondo de la imagen cercanos a personas erróneamente clasificados como persona [31].

Solo hay un algoritmo en el Estado del Arte de segmentación persona-fondo [31], por lo que el resto de la sección se dedicará a la explicación del mismo y no a la clasificación de algoritmos disponibles, tal y como se ha hecho en la sección anterior.

El método propuesto en [31], parte del algoritmo implementado en [28] para obtener mapas de confianza de detección de partes del cuerpo y los combina y extiende apropiadamente para obtener un único mapa de confianza acorde con las partes del cuerpo detectadas. Finalmente, se genera la máscara de segmentación de fondo como resultado de la binarización del mapa de confianza y posterior post-procesado de la imagen binaria empleando operadores morfológicos.

Se presentan en total 5 maneras de emplear la detección de las distintas partes del cuerpo para realizar la segmentación: **IBP** (*Independent Body Parts*), **DBP** (*Dependent Body Parts*), la versión extendida de ambas, **IEBP** y **DEBP** (*Independent* y *Dependent Extended Body Parts* respectivamente), y la versión con post-procesado **DEBP-P** (*Dependent Extended Body Parts Post-Processed*). A continuación, se explica cada una de ellas.

El detector basado en detección de partes y multiescala (ver figura 2.3 (a)) define la confianza de cada píxel (x, y) como $P_n(x, y, s)$ para cada parte del cuerpo n $(n = 1, \ldots, N)$ asociada a la escala s $(s = 1, \ldots, S)$. Cada parte del cuerpo se define completamente mediante tres elementos $(F_n, v_{n,0}, d_n)$ donde $F_n$es el la respuesta obtenida del filtrado **HOG** (*Histogram of Oriented Gradients*) para cada parte del cuerpo n; $v_{n,0}$ es un vector

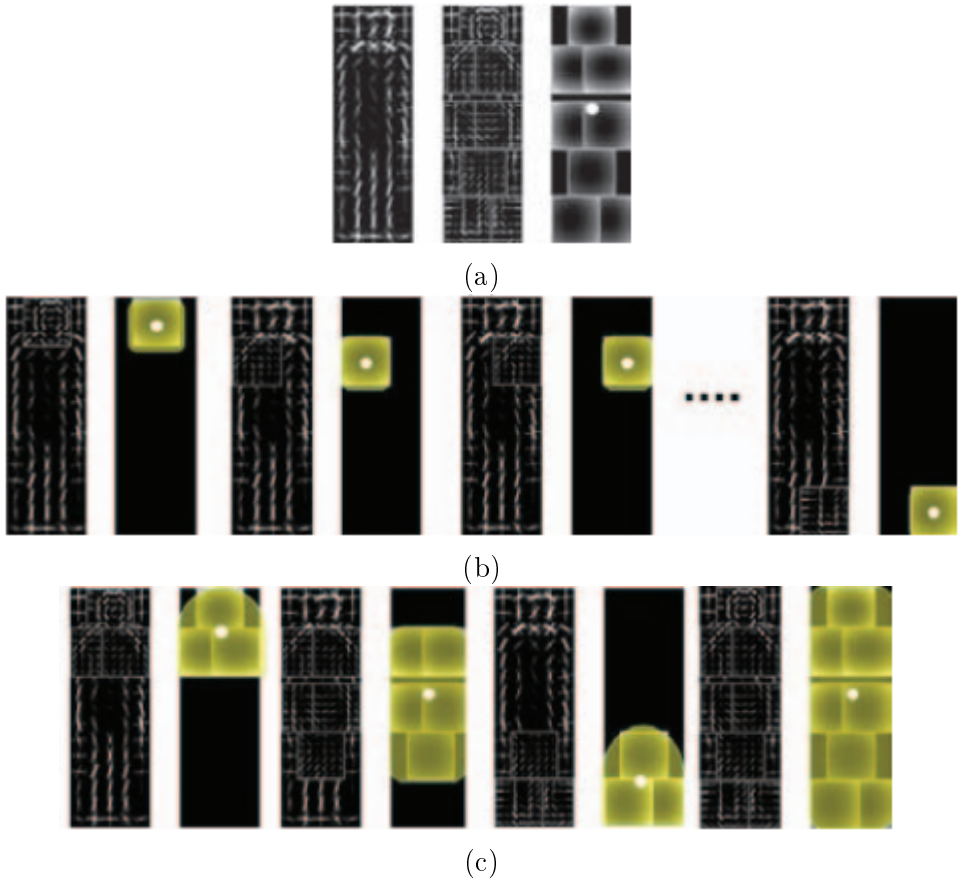

Figura 2.3: Representación de modelos de partes del cuerpo. (a) Modelo de personas propuesto en [28]. (b) Modelo independiente (**IBP**). (c) Modelo dependiente (**DBP**). Fuente: [31].

bidimensional que define la posición relativa de la parte n con respecto a la posición del cuerpo completo (posición de anclaje (x_0, y_0)); y d_n es un vector de cuatro dimensiones que especifica los coeficientes de la función cuadrática que define los costes para cada posible posición de la parte con respecto a la posición de anclaje. Por tanto, el valor de confianza para cada píxel, $P_n(x, y, s)$ se puede obtener restando F_n menos la deformación definida por d_n.

En el método **IBP** se emplean ocho partes del cuerpo distintas y la posición de anclaje de cada parte se define respecto de la propia parte (ver figura 2.3 (b)). Para aportar mayor robustez a la detección se emplea el método **DBP** que utilizan M modelos de partes dependientes D_m, con $m = 1, \ldots, M$ como combinaciones de partes independientes (figura 2.3 (c)) cuya posición de anclaje es relativa a la correspondiente parte del cuerpo dependiente D_m. Se han empleado $M = 4$ partes dependientes: cabeza y hombros, tronco, piernas y cuerpo entero, lo cual permite explotar la correlación entre las partes del cuerpo. Con el objetivo de recuperar partes dependientes no detectadas y normalizar la confianza de la detección entre partes dependientes ya detectadas, se propone extender la definición de la partes dependientes y reutilizar la información de otras partes. La nueva parte dependiente D'_m viene dada por el máximo entre la parte original dependiente D_m y la media de otras partes dependientes, relativas al mismo D_m. De esta manera, si solo se observan dos partes dependientes, el resto de partes se pueden recuperar y normalizar realizando una media de todas las demás.

Una vez obtenidos los mapas de confianza de cada parte, independiente o dependiente, se pueden extender y dar lugar a los métodos **IEBP** y **DEBP** respectivamente. **IEBP** extiende cada parte del cuerpo independiente, mientras que **DEBP** extiende cada combinación de partes del cuerpo dependiente. Ambos, **IEBP** y **DEBP** cubren la parte detectada en la representación elegida de dicha parte como se muestra en la figura 2.3 (b) y (c) en amarillo, de acuerdo al área que se supone deben cubrir en la imagen. Finalmente se elige el mayor valor de confianza de todas las escalas $C(x, y)$.

Por último, para obtener la máscara del fondo del la imagen, se binariza $C(x, y)$ y se eliminan las regiones que son más pequeñas que el tamaño mínimo de una persona definido en [28]. Para ello, la máscara resultante se erosiona con un disco del tamaño de la parte más pequeña del cuerpo a detectar en el tamaño mínimo de una persona, y a continuación se realiza un análisis de las componentes conexas para eliminar las regiones

más pequeñas que el tamaño mínimo de persona. Finalmente, se dilata la imagen con un disco del tamaño de la parte más pequeña del cuerpo a detectar en el tamaño máximo de una persona. A este método se le denomina **DEBP-P**.

3

Algoritmo

3.1. Introducción

En este capítulo, se explicará de manera exhaustiva el algoritmo de estimación de densidad de personas implementado en este proyecto, basado en el método propuesto por Ryan en [83]. Se ha elegido este estimador de densidad del Estado del Arte por dos razones principales. La primera de ellas, es que como se ha explicado en la sección 2.2.2 del capítulo anterior, este algoritmo de estimación de densidad se basa en la extracción del fondo de la imagen, lo cual permitirá comparar los resultados obtenidos al utilizar segmentación persona-fondo y segmentación frente-fondo. Por otro lado, según los resultados presentados en [83], el método propuesto ofrece muy buenos resultados cuando se compara con otros algoritmos similares del Estado del Arte.

El algoritmo de estimación de densidad de personas implementado incluye las siguientes etapas:

1. Segmentación: En esta etapa, se segmenta la imagen obteniendo una máscara sólo con los píxeles del frente.

2. Extracción de características: Se extraen las características elegidas de los segmen-

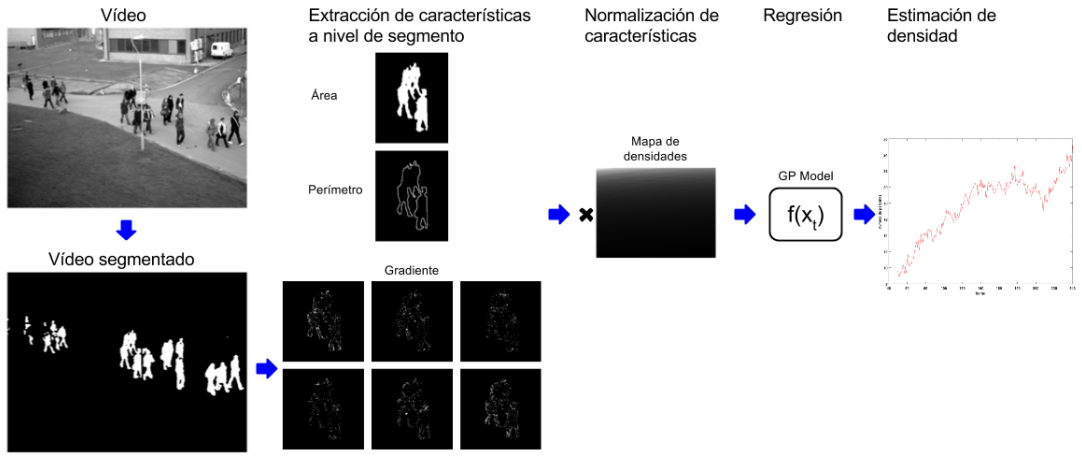

Figura 3.1: Diagrama del algoritmo de estimación de densidad de personas.

tos del frente de la imagen.

3. Normalización de características: Es necesario normalizar las características extraídas con el objetivo de corregir los efectos de la perspectiva y la distorsión geométrica debida a la configuración de la cámara, tal y como se explica en la sección 2.2.2 del capítulo de Estado del Arte.

4. Regresión: La etapa de regresión, permite estimar el número de personas de la imagen para un vector de características dado. Para ello, es necesario un proceso previo de entrenamiento en el que se determina la relación entre el *ground truth* y las características, que permitirá inferir el número de personas durante la estimación. El algoritmo de regresión empleado en este caso se denomina proceso gaussiano.

En la figura 3.1 se muestra un esquema con las distintas etapas que constituyen el algoritmo de estimación de densidad.

En las siguientes secciones, 3.2, 3.3, 3.4 y 3.5, se explica cada una de estas etapas.

3.2. Segmentación

La segmentación es una etapa fundamental del algoritmo, ya que permite extraer aquellas partes de la imagen que contienen a las personas y que por tanto, son relevantes para la estimación. Si no se realizase este paso, sería mucho más difícil establecer una relación

entre las características extraídas (de toda la imagen y no solo de los segmentos) y el número de personas. Por este motivo, los métodos basados en detección de personas utilizan características mucho más complejas y propias de personas.

Como ya se ha comentado antes, uno de los objetivos de este proyecto es comparar los resultados de la estimación utilizando dos tipos de segmentadores: segmentador persona-fondo y segmentador frente-fondo. Para ello, se ha implementado el algoritmo de estimación de densidad utilizando un segmentador de cada tipo. En las siguientes secciones, 3.2.1 y 3.2.2, se hace referencia a los algoritmos utilizados en cada caso.

3.2.1. Segmentación frente-fondo

El algoritmo de segmentación frente-fondo utilizado en este proyecto es un algoritmo de extracción de fondo (**BGS**), como bien se ha explicado ya en la sección 2.3.1 del capítulo anterior. Concretamente, se ha empleado el algoritmo **SGMM-SOD** [25, 26], desarrollado por Rubén Heras Evangelio, proporcionado por el propio autor y por la *Technical University of Berlin*. La elección de este algoritmo se debe a que ha sido reconocido por *Change Detection*[1], web especializada en la evaluación de este tipo de algoritmos, como uno de los mejores, de acuerdo a los resultados que ofrece.

Este algoritmo utiliza dos modelos de fondo distintos para la detección de objetos estáticos y en movimiento en secuencias de video con gran densidad de personas. Uno de los modelos está dedicado a la detección de movimiento precisa, mientras que el otro busca conseguir una representación de la escena vacía. Las diferencias en la detección del frente de la imagen de los modelos complementarios se utilizan para identificar nuevas regiones estáticas. Posteriormente, se realiza un análisis de las regiones detectadas para comprobar si hay un nuevo objeto en la escena o si algún objeto ya no está en la misma. Se evita que los objetos estáticos sean incorporados en el modelo de escena vacía y se eliminan rápidamente los objetos que ya no están en la escena de ambos modelos. De esta manera, se consigue un modelo muy preciso de la escena vacía y se obtienen mejores resultados que en la segmentación con un único modelo de fondo.

Para realizar la segmentación se realiza un análisis de cada *frame* de la escena en dos niveles distintos:

[1] http://changedetection.net/

- **A nivel de píxel:** Los píxeles se clasifican de acuerdo a los resultados obtenidos mediante la substracción de ambos modelos de fondo.
- **A nivel de región:** Las nuevas regiones estáticas se clasifican como objetos estáticos u objetos eliminados de la escena.

Los resultados de la clasificación a nivel de región se emplean también en el análisis a nivel de píxel para conseguir una buena representación de la escena vacía. Esto permite la inicialización del modelo de escena vacía de manera implícita, sin necesidad de tener un período de inicialización.

Como bien se ha comentado ya, este algoritmo utiliza dos modelos complementarios de tipo **GMM** (*Gaussian Mixture Model*), uno, B_S, dedicado a segmentar movimiento a corto plazo y otro, B_L, dedicado a la reconstrucción de la escena vacía a largo plazo. Ambos modelos han sido configurados con los mismos parámetros a excepción de la tasa de aprendizaje. Por un lado, el modelo a largo plazo se adapta rápidamente a cambios en la escena tales como cambios de iluminación y la incorporación o eliminación de objetos estáticos. Por otro lado, El modelo a largo plazo tiene una tasa de aprendizaje menor y por tanto reacciona menos a los cambios en la escena. Cuando el modelo B_L detecta un píxel del frente en el tiempo t, que no es detectado por el modelo B_S, significa que se ha encontrado una nueva descripción estable del sistema y que hay un nuevo modo en el fondo de B_S. En este caso, se propaga B_S a B_L, en B_L se elimina el nuevo modo y se deja de aprender, manteniendo así una copia del fondo de la escena como en el tiempo $t-1$ y dejando que B_S aprenda la nueva descripción. Cuando un píxel del frente de la imagen es detectado por B_S pero no por B_L, significa que el fondo de la escena ha sido detectado, por lo que se propaga B_L a B_S restableciendo así el modelo de fondo.

Dado que B_L y B_S pueden contener descripciones diferentes de la escena estática, la detección del frente, obtenida mediante la substracción de ambos modelos, puede ser también diferente. Concretamente, B_S es capaz de incorporar nuevas descripciones de la escena estática a nivel de píxel mientras que B_L no. Este hecho se utiliza para detectar píxeles que describen una nueva apariencia de la escena estática (nuevos píxeles estáticos del frente). Para ello, se ha empleado una máquina de estados que permite hacer dicha clasificación y controlar las acciones correspondientes sobre el modelo de fondo (ver diseño de la máquina de estados en [26]).

Los nuevos píxeles estáticos del frente de la imagen se agrupan en regiones, que pueden corresponder a nuevos objetos estáticos o a objetos eliminados. Estas nuevas regiones tardan en formarse completamente ya que no todos los píxeles son ocluidos o descubiertos a la vez. El análisis de las regiones solo se puede hacer una vez que la región está completamente formada, es decir, cuando deja de crecer. Por este motivo, se mantiene una lista de las nuevas regiones estáticas encontradas en los *frames* recientes y se guarda información sobre la caja que rodea a dicha región (posición y tamaño), la primera y la última vez que se detectó y su clasificación (nuevo objeto estático, objeto estático eliminado o sin determinar aún). La región se analiza una vez formada y se buscan sus bordes en el *frame* de entrada y en B_L. Si los bordes de la máscara de la región coinciden con los encontrados en B_L, significa que se trata de un objeto eliminado y si coinciden con los del *frame* de entrada, significa que se trata de un nuevo objeto estático. Tan pronto se clasifican las regiones se actualizan los modelos de fondo.

Las regiones clasificadas como objetos eliminados descubren una región de la escena vacía que no era visible hasta ese momento, por lo que se añaden al modelo B_L mediante el envío de la posición y el tamaño de la región al nivel de píxel. Por otro lado, los píxeles de las nuevas regiones estáticas se propagan de B_S a B_L. Por tanto, el modelo de la escena vacía se actualiza a lo largo de toda la secuencia y no sólo al comienzo con la inicialización del modelo de fondo.

El algoritmo proporcionado está implementado en C++ y proporciona directamente las máscaras de los *frames* para la secuencia de entrada y no se ha modificado ningún parámetro de configuración para su ejecución. En la figura 3.2 se muestra un ejemplo de la máscara obtenida utilizando este algoritmo.

3.2.2. Segmentación persona-fondo

El algoritmo de segmentación persona-fondo utilizado en este proyecto es el algoritmo descrito en [31] y ha sido facilitado por el **VPULab**. Como ya se comentó en el capítulo anterior, el algoritmo cuenta con 5 versiones que se corresponden con 5 maneras distintas de emplear la detección de las distintas partes del cuerpo:

1. **IBP** (*Independent Body Parts*): Se analizan 8 partes del cuerpo por separado.

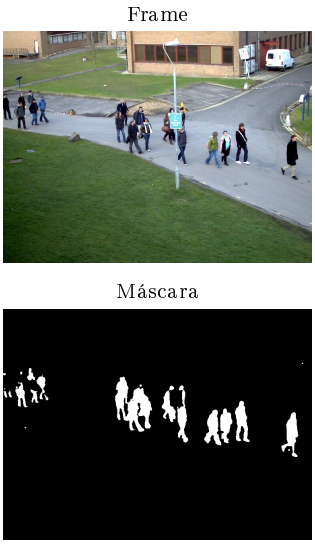

Figura 3.2: Ejemplo de la segmentación frente-fondo.

2. **DBP** (*Dependent Body Parts*): Se definen 4 partes del cuerpo dependientes como combinación de otras partes independientes: Cabeza y hombros, tronco, piernas y cuerpo entero.

3. **IEBP** (*Independent Extended Body Parts*): Se extiende el área que ocupa cada parte del cuerpo independiente.

4. **DEBP** (*Dependent Extended Body Parts*): Se extiende el área que ocupa cada parte del cuerpo dependiente.

5. **DEBP-P** (*Dependent Extended Body Parts Post-Processed*): Se realiza un post procesado para mejorar la máscara de salida.

En este proyecto se utilizarán las versiones **DEBP** y **DEBP-P** del algoritmo, ya que ofrecen mejores resultados atendiendo a los datos presentados en [31].

A continuación se explicará de manera práctica el funcionamiento de los métodos **DEBP** y **DEBP-P**.

En primero lugar, se crea una máscara de procesamiento en función de los parámetros de ejecución del algoritmo y se obtiene el vector F_n como respuesta del filtrado **HOG** (*Histogram of Oriented Gradients*) para cada parte del cuerpo n. Cada parte del cuerpo se define completamente mediante tres elementos, $(F_n, v_{n,0}, d_n)$, donde $v_{n,0}$ es un vector

bidimensional que define la posición relativa de la parte n con respecto a la posición del cuerpo completo (posición de anclaje (x_0, y_0)); y d_n es un vector de cuatro dimensiones que especifica los coeficientes de la función cuadrática que define los costes para cada posible posición de la parte con respecto a la posición de anclaje.

A continuación, se calcula la confianza de cada píxel (x, y) como $P_n(x, y, s)$ para cada parte del cuerpo n asociada a la escala s ($s = 1, \ldots, S$) y se obtiene el mapa de confianza utilizando la siguiente expresión:

$$P_n(x, y, s) = F_n(x, y, s) - \langle d_n, \phi\,(dx_n, dy_n)\rangle \tag{3.1}$$

siendo

$$(dx_n, dy_n) = (x_n, y_n) - (2(x_0, y_0) + v_{n,0}) \tag{3.2}$$

dado el desplazamiento de la parte n relativo a la posición de anclaje y

$$\phi(dx, dy) = (dx, dy, dx^2, dy^2) \tag{3.3}$$

que define las distribuciones potenciales de deformación espacial.

Además de la detección de cada una de las 4 partes dependientes, se realiza la detección de cada parte dependiente basada en las 3 partes restantes para recuperar partes no detectadas. Las detecciones se combinan mediante el cálculo del máximo entre la parte dependiente original, D_n y la media entre las otras partes dependientes relativas a D_n:

$$D'_n(x, y, s) = max\left(D_n(x, y, s), \frac{1}{N-1}\sum_{1\neq n}^{N} D(x, y, s)\right) \tag{3.4}$$

Una vez realizadas las detecciones de las 4 partes, se extienden los mapas de confianza de cada una de las partes y se elige el mayor valor de confianza de todas las escalas. Finalmente, se combinan los mapas de confianza de cada una de las partes dependientes, se umbralizan y se obtiene una máscara binaria. En la versión **DEBP-P** se eliminan las regiones más pequeñas que el tamaño mínimo de una persona definido en [28] y se dilata

la imagen binaria con un disco del tamaño de la parte más pequeña del cuerpo a detectar en el tamaño máximo de una persona.

Antes de la ejecución del algoritmo se deben fijar 3 parámetros básicos utilizados por el detector [28]:

1. El tamaño del modelo (*model.sbin*): Este parámetro se debe fijar en dependencia del tamaño de las personas dentro de la secuencia, ya que determina el tamaño de las partes que se busca en cada imagen. Los valores habituales son 8 y 4, siendo 8 el valor que se debe elegir cuando las personas son grandes y 4 el valor a elegir cuando son pequeñas.

2. Escala (*scale_levels*): Este parámetro permite escalar el tamaño definido por el *sbin* cuando no todas las personas en la secuencia son aproximadamente del mismo tamaño.

3. Umbral (*thresholds*): determina el umbral que se aplica a la confianza de detección.

Para la utilización de este algoritmo se han fijado los parámetros 1 y 2 de acuerdo con características de la secuencia en cuestión y se han extraído las máscaras binarias, tanto para **DEBP** como para **DEBP-P**, para varios umbrales con el fin de encontrar los mejores resultados. En la figura 3.3 se muestran un ejemplo de las las máscaras obtenidas para los umbrales 0.70:0.05:0.90 y las dos versiones: con post-procesado y sin post-procesado.

3.3. Extracción de características

A continuación, siguiendo con el esquema presentado en la sección 3.1 figura 3.1, se realiza la extracción de características. La extracción de características para la estimación de densidad se realiza sobre cada segmento, es decir, a nivel local, lo que permite estimar el número de personas en cada grupo. Las características empleadas son área, perímetro e histograma de orientación de gradientes, al igual que en [83]. A continuación, se explicará cómo se ha calculado cada una de ellas.

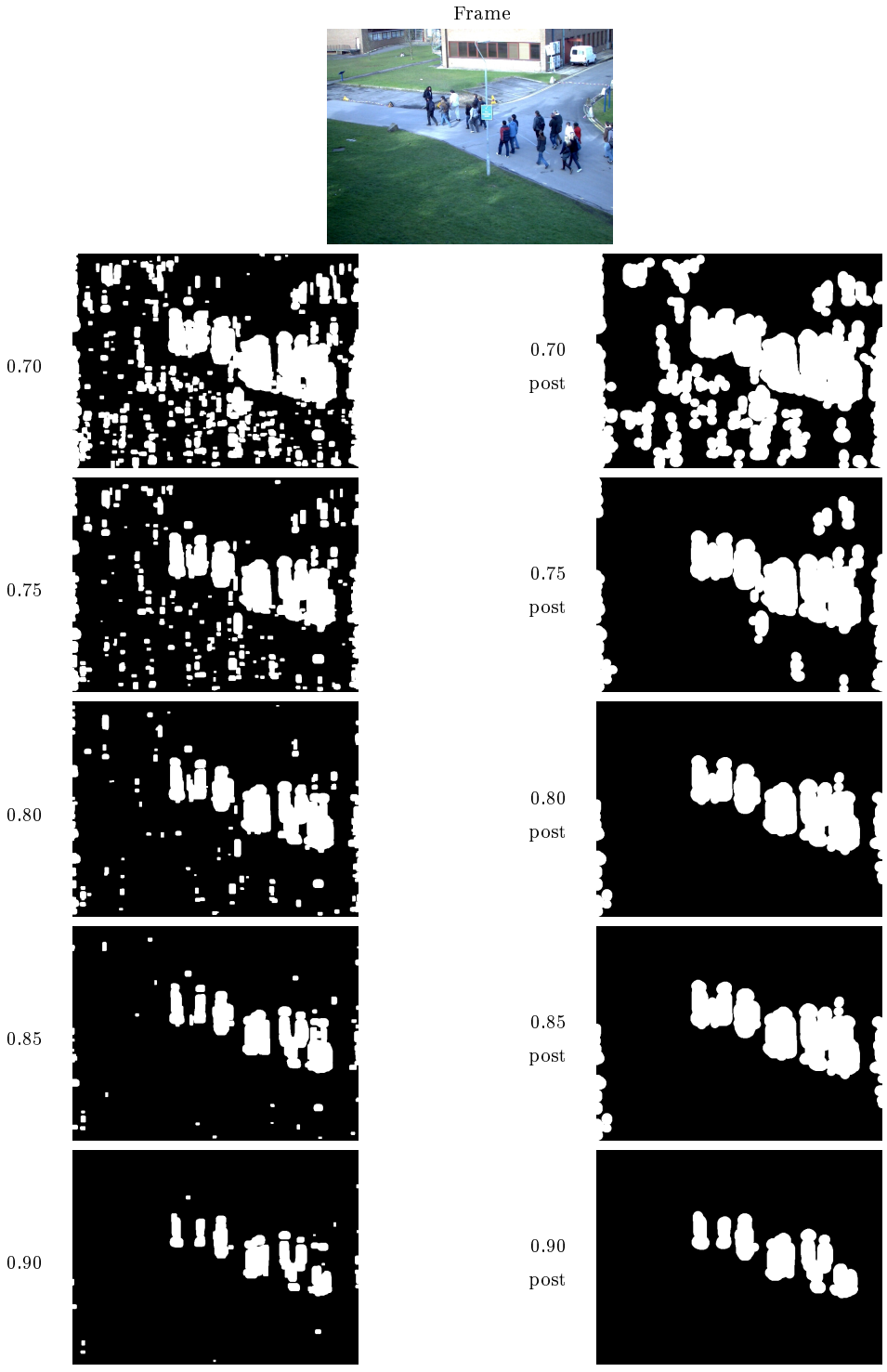

Figura 3.3: Ejemplo de la segmentación persona-fondo para distintos umbrales con y sin post-procesado.

- **Área**

  El área se calcula como el número de píxeles D dentro de cada segmento B_n.

$$A_n = \sum_{(i,j)\epsilon B_n} D(i,j) \tag{3.5}$$

- **Perímetro**

  El perímetro se calcula como el número de píxeles P dentro del perímetro de cada segmento P_n.

$$L_n = \sum_{(i,j)\epsilon P_n} D(i,j) \tag{3.6}$$

- **Histograma de orientación de gradientes**

  Para calcular el histograma de orientación de gradientes, se utiliza el operador de Sobel. Para ello, se deben calcular en primer lugar las derivadas horizontal y vertical como se muestra a continuación:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \tag{3.7}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \tag{3.8}$$

  Donde $*$ representa la convolución entre los *kernels* horizontal y vertical de Sobel y la imagen I en escala de grises. Una vez calculadas las derivadas, se obtienen la magnitud y la orientación del gradiente:

$$|G(i,j)| = \sqrt{G_x(i,j)^2 + G_y(i,j)^2} \tag{3.9}$$

$$\angle G(i,j) = arctan\left(\frac{G_y(i,j)}{G_x(i,j)}\right) \tag{3.10}$$

  El histograma de orientación de gradientes para cada segmento, E_n, se construye a partir de la magnitud del gradiente para $H = 6$ rangos de orientación distintos

entre 0^{o} y 180^{o}. Por tanto, la contribución de cada píxel de un segmento, $(i,j)\epsilon B_n$, será proporcional a la magnitud del gradiente, $|G(i,j)|$.

Siendo el valor de cada barra del histograma $E_n[h]$, y el límite inferior del rango de orientación para dicha barra θ_h, el histograma de orientación de gradientes se define como:

$$E_n[h] = \sum_{(i,j)\epsilon B_n} \begin{cases} |G(i,j)| & si\ \theta_h \leq \angle G(i,j) < \theta_{h+1} \\ 0 & en\ cualquier\ otro\ caso \end{cases} \tag{3.11}$$

La utilización del histograma de orientación de gradientes ayuda a diferenciar a las personas de otros objetos en la escena y a identificar oclusiones cuando un grupo de personas se ocluyen parcialmente entre sí. Mientras que el área y el perímetro de los segmentos se reducen cuando hay oclusiones, los gradientes de la imagen son más intensos debido al solapamiento de partes del cuerpo de las personas, a la diferencia en los tonos de la piel, o a variaciones en la ropa.

Para el proceso de regresión (tanto en el test como en el entrenamiento), se construye un único vector de características que contiene todas las características extraídas para el segmento $n-ésimo$:

$$x_n = [A_n, L_n, E_n[1], \ldots, E_n[H]] \tag{3.12}$$

Para realizar la extracción de características como se ha descrito en esta sección, se han implementado en Matlab los scripts y funciones necesarios que para cada frame y las máscara de segmentación correspondiente de entrada, proporcionan el vector de características x_n de salida.

3.4. Normalización de características

Tal y como se muestra en el esquema presentado en la sección 3.1 figura 3.1, después de la extracción de características se lleva a cabo su normalización. Como bien se ha explicado ya en la sección 2.2.2 del capítulo 2, la normalización de las características extraídas se realiza principalmente por dos motivos:

1. Corregir el efecto de la perspectiva: objetos cercanos a la cámara ocupan muchos píxeles en comparación con objetos del mismo tamaño más alejados de la cámara. Además, el ángulo de observación de la cámara a un objeto con respecto al plano de tierra no es constante dentro de la imagen.

2. Corregir la distorsión geométrica debida al cambio de configuración de la cámara: es deseable que las características sean, no solo invariantes a traslaciones de los peatones, sino también, al punto de vista de la cámara.

Hay en la literatura muchos ejemplos de normalización de características para corregir todos o algunos de los problemas antes citados. En este proyecto se utilizarán dos tipos de correcciones diferentes dependiendo de la base de datos empleada y los datos disponibles para la misma.

- **Calibración de la cámara**

  A través de la calibración de la cámara se pueden normalizar las características extraídas y corregir, tanto los efectos de la perspectiva, como el cambio de configuración de la cámara. Se han descrito hasta el momento varios métodos de calibración [92, 103], aunque el método más popular de todos es el modelo de Tsai [92], utilizado habitualmente en muchas bases de datos de vídeo-vigilancia. El modelo de Tsai incorpora parámetros como la posición de la cámara, el ángulo de rotación, la distancia focal y la distorsión radial de la lente, para mapear las coordenadas del mundo real en 3D (x, y, z) a las coordenadas del plano de la imagen en 2D (i, j). Estos parámetros se estiman a partir de un set de correspondencias de puntos especificados manualmente entre los píxeles de la imagen y su localización en el mundo real en el plano de tierra. Una gran ventaja del método de Tsai es que estos parámetros están ya disponibles para muchas de las bases de datos.

  Para normalizar las características, se ha elaborado un mapa de densidades que consiste en la asignación de un peso a cada píxel de la imagen, de tal manera que se compensen los efectos de la perspectiva. Con el objetivo de calcular el mapa de densidades, se ha utilizado un cilindro de radio $r = 0{,}25$ metros y altura $h = 1{,}7$ metros, que modela la forma de las personas, y se ha proyectado sobre todos los píxeles de la imagen (i, j). El área del cilindro proyectado sobre la imagen se denota como $S(i, j)$ (ver figura 3.4). Finalmente, el peso que se asigna a cada píxel, $D_2(i, j)$,

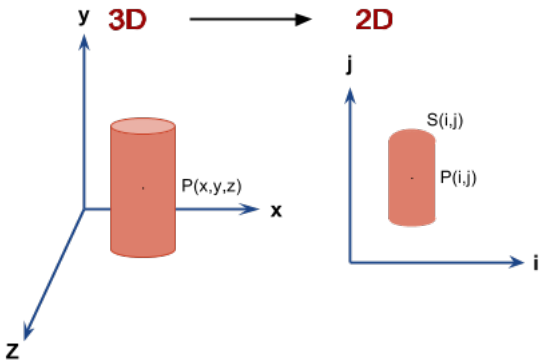

Figura 3.4: Proyección de modelo de persona de 3D a 2D.

se calcula como el inverso del área de un objeto proyectado y centrado en dicho píxel.

$$D_2(i,j) = \frac{1}{S(i,j)} \tag{3.13}$$

En la figura 3.5 se muestra un ejemplo de cómo serían los cilindros proyectados sobre una imagen real.

El hecho de que el modelo cilíndrico de personas, no represente de manera precisa la forma o el tamaño de las personas, no es importante, ya que su verdadero objetivo es normalizar las características.

El mapa de densidades D_2 asigna a cada píxel un peso, de manera que un grupo de píxeles, B, tenga un área $\sum_{(i,j)\epsilon B} D_2(i,j)$. Esto hará que los objetos que se encuentren lejos de la cámara y que por tanto, ocupen menos píxeles, sean compensados con pesos mayores en el mapa de densidades. Además, D_2 se podrá escalar fácilmente para diferentes ángulos de observación de la cámara y las características normalizadas serán invariantes a la escena en cuestión.

El mapa de densidades D_2 es adecuado para las características bidimensionales como el área, pero para las características de una sola dimensión como son el perímetro y los bordes, se calcula otro mapa de densidades D_1 como la raíz cuadrada del anterior.

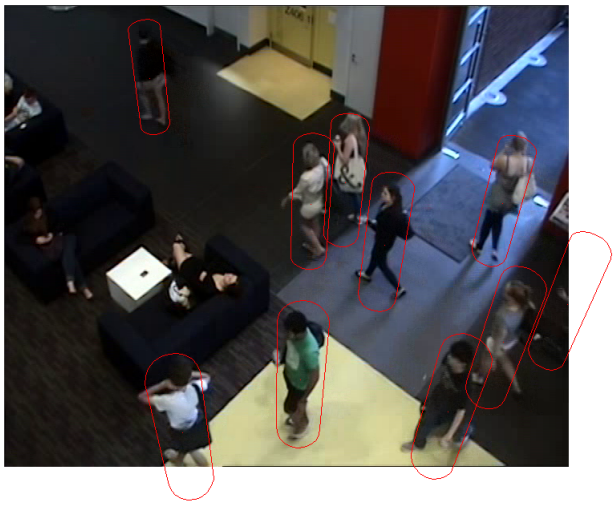

Figura 3.5: Ejemplo de una imagen real con proyección de modelos de personas cilíndricos.

$$D_1(i,j) = \sqrt{D_2(i,j)} = \frac{1}{\sqrt{S(i,j)}} \tag{3.14}$$

Para obtener las proyecciones de los puntos de 3D a 2D y viceversa, se ha utilizado la herramienta en C++ accesible desde la web de PETS 2006[2], cortesía del proyecto **ETISEO**, que permite cargar los parámetros de calibración, disponibles junto a la base de datos en un fichero XML, y calcular dichas proyecciones. Esto se ha complementado con un script de Matlab para elaborar los mapas de densidades D_1 y D_2.

Como se ha comentado antes, este método de normalización es el más completo, ya que permite, no sólo la corrección de los efectos de perspectiva, sino también, obtener características invariantes a la escena. Sin embargo, solo se podrá utilizar este método cuando se proporcionen los parámetros de calibración para la bases de datos en cuestión.

[2]http://www.cvg.reading.ac.uk/PETS2006/data.html

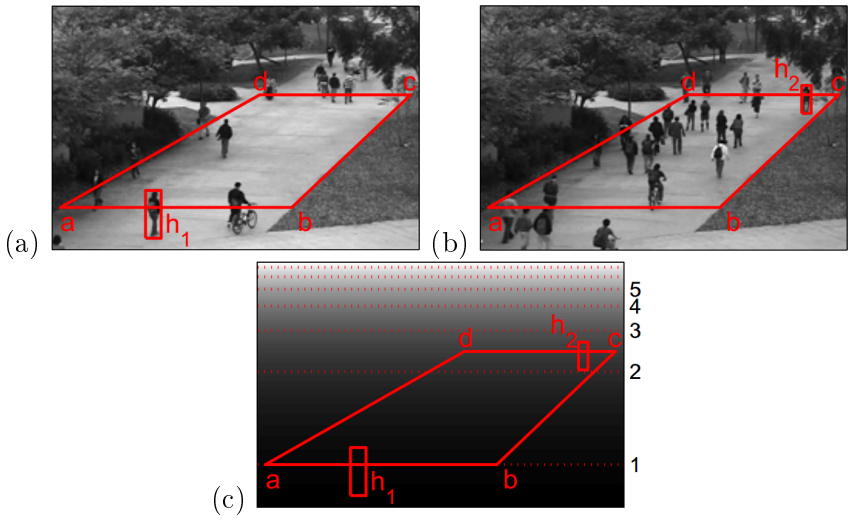

Figura 3.6: Mapa de perspectiva: (a) Persona de referencia en el extremo inferior de la escena. (b) Persona de referencia en el extremo superior de la escena. (c) Mapa de perspectiva con los píxeles escalados según su tamaño relativo en la escena 3D real. Fuente: [11].

- **Normalización de perspectiva mediante píxel de referencia**

  Si la calibración no está disponible se puede recurrir a otros métodos para realizar la corrección de perspectiva como por ejemplo, el propuesto en [11]. En este método se elabora un mapa de perspectiva mediante la interpolación lineal entre dos extremos de la escena. En primer lugar se marca un plano de tierra como en la figura 3.6 (a) y se miden las distancias $|\overline{ab}|$ y $|\overline{cd}|$. A continuación, se selecciona una persona como referencia y se miden sus altura, h_1 y h_2, cuando su centro se encuentra en $\overline{ab}$ y $\overline{cd}$ respectivamente (ver figuras 3.6 (a) y 3.6 (b)). A los píxeles en $\overline{ab}$ se les asigna un peso de 1 y a los píxeles en $\overline{cd}$ se les asigna un peso de $\frac{h_1|\overline{ab}|}{h_2|\overline{cd}|}$. Por último, se calculan los píxeles restantes mediante la interpolación lineal entre ambas líneas. La figura 3.6 (c) muestra el mapa de perspectiva de las escena completa. El mapa obtenido se denomina D_2 y se aplica sobre las características de área, mientras que para las características de perímetro y bordes, se utiliza la raíz cuadrada del anterior, $D_1(i,j) = \sqrt{D_2(i,j)}$, igual que se hacía en el método anterior.

  A diferencia del mapa de densidades descrito en el punto anterior, el mapa de perspectiva se ha proporcionado junto al *dataset* empleado, por lo que no se ha implementado ningún script para calcularlo.

Una vez calculados los mapas de densidades D_1 y D_2 para las características de una y dos dimensiones respectivamente (siguiendo con el esquema presentado en la figura 3.1 de la sección 3.1), se deben aplicar a las características descritas en las sección anterior 3.3 tal y como se muestra a continuación:

- **Área:** la ecuación 3.5 quedaría como:

$$A_n = \sum_{(i,j)\epsilon B_n} D_2(i,j) \tag{3.15}$$

- **Perímetro:** la ecuación 3.6 quedaría como:

$$L_n = \sum_{(i,j)\epsilon P_n} D_1(i,j) \tag{3.16}$$

- **Histograma de orientación de gradientes:** la ecuación 3.7 se transforma tal y como se muestra a continuación:

$$E_n[h] = \sum_{(i,j)\epsilon B_n} \begin{cases} D_1(i,j) \times |G(i,j)| & si\ \theta_h \leq \angle G(i,j) < \theta_{h+1} \\ 0 & en\ cualquier\ otro\ caso \end{cases} \tag{3.17}$$

3.5. Regresión

La etapa final del algoritmo de estimación de densidad de personas como se muestra en el esquema de la sección 3.1 figura 3.1 es la regresión. Como bien se ha comentado ya, la regresión permite inferir el número de personas en la escena a partir de las características de entrada. Para ello, es necesaria una primera fase de entrenamiento en la que se determina la función de regresión que establece la relación entre las características y el número de personas a partir de un set de imágenes de entrenamiento. Una vez conocida dicha función, es posible estimar la densidad a partir de las características extraídas.

En la siguiente sección, 3.5.1, se explicará cómo se han obtenido los vectores de *ground truth* para realizar el entrenamiento y el test. A continuación, en la sección 3.5.2, se realizará un resumen del algoritmo de regresión utilizado, denominado proceso gaussiano.

3.5.1. Ground truth

El primer paso para realizar el entrenamiento es definir el set de imágenes de entrenamiento y obtener el *ground truth* que, junto con el vector de características, x_n, definido en la sección 3.3, conformarán los datos de entrada del algoritmo de entrenamiento.

Como el algoritmo implementado en este proyecto calcula el número de personas como la suma total del número de personas en cada segmento, el entrenamiento se debe realizar también a nivel local y por tanto, el *ground truth* debe especificar el número de personas para cada segmento.

El *ground truth* para el entrenamiento se calcula de manera automática a partir de la anotación del píxel central de las personas (*dot annotation*), disponible para todas las bases de datos utilizadas en este proyecto.

Se desea que el *ground truth* sea completamente independiente del proceso de segmentación para realizar el entrenamiento, de manera que se pueda aprender lo mejor posible la relación entre el número de personas dentro de cada segmento (*ground truth*) y las características extraídas para dicho segmento. En este sentido, se considera el hecho de que una persona pueda estar dividida en más de un segmento del frente de la imagen, ante lo cual, sería deseable asignar una parte proporcional de la persona a cada segmento. Además de este tipo de errores, hay otras situaciones que pueden afectar al proceso de entrenamiento, como por ejemplo, que una persona esté entrando o saliendo de la región de interés (**ROI**) y por tanto, parte de su cuerpo esté fuera de la misma. En este caso, conviene también contar sólo una fracción de la persona en el *ground truth*. Para lidiar con ambas situaciones, en el cálculo del *ground truth* se ha utilizado un modelo cilíndrico que permite definir el contorno de las personas y asignar a los segmentos fracciones de estas. Estos cilindros se corresponderán con los cilindros proyectados de la sección 3.4 obtenidos para la normalización de características en aquellas secuencias en las que se disponga de los parámetros de calibración. En caso contrario, se definirán simplemente cajas rectangulares alrededor de las personas. En la figura 3.7 se puede ver como se han definido los modelos cilíndricos de las personas y las cajas en cada caso.

A continuación, se presenta detalladamente cómo se han realizado los cálculos, para lo cual, en la tabla 3.1 se define previamente la notación utilizada.

(a) (b)

Figura 3.7: Modelos cilíndricos de personas para el cálculo del *ground truth*. (a) Modelos cilíndricos proyectados de 3D a 2D para la base de datos PET 2009, obtenidos utilizando los parámetros de calibración disponibles. (b) Cajas rectangulares definidas para la base de datos UCSD ya que no se dispone de parámetros de calibración.

Notación	**Descripción**
M	Región de interés (**ROI**).
F	Píxeles del frente de la imagen.
B	Píxeles del frente de la imagen dentro de la **ROI** ($B = M \cap F$).
B_n	Segmento n dentro de B.
R_i	Contorno de la persona i.
$R_i \cap B_n$	Píxeles del frente de la imagen dentro del contorno R_i pertenecientes al segmento B_n.
$R_i \cap B$	Píxeles del frente de la imagen dentro del contorno R_i y dentro de la **ROI**.

Cuadro 3.1: Notación utilizada para el cálculo del *ground truth*.

La parte de la persona $i-ésima$ dentro de la región de interés se define como:

$$Q_i = \frac{|M \cap R_i|}{|R_i|} \tag{3.18}$$

La contribución de la persona $i-ésima$ al segmento n se define como:

$$C_{in} = \frac{|R_i \cap B_n|}{|R_i \cap B|} \times Q_i \tag{3.19}$$

Por tanto, el número total de personas dentro del segmento n (vector de *ground truth*) sería:

$$f_n = \sum_i C_{in} \tag{3.20}$$

Para evaluar los resultados obtenidos en la etapa de test, se debe obtener el *ground truth* holístico de cada una de las imágenes del set de test. Para ello, simplemente se cuenta el número de centros o puntos de personas (*dot annotation*) que se encuentran dentro de la región de interés (**ROI**). Siendo M la región de interés y D_i el centro de la $i-ésima$ persona, el *ground truth* de test para cada imagen se calcularía de la siguiente manera:

$$f = \sum_i |M \cap D_i| \tag{3.21}$$

Para realizar el cálculo del *ground truth* en todas las secuencias utilizadas de la manera que se ha contado en esta sección, se han implementado en Matlab todos los scripts y funciones necesarios.

3.5.2. Proceso gaussiano

El algoritmo de regresión empleado en este proyecto se denomina proceso gaussiano (**GP**) [79, 80]. En esta sección, se explicará cómo formular un sistema bayesiano para realizar la regresión basada en un proceso gaussiano. La ventaja principal, por la cual se ha elegido este método de regresión, es que no hace ninguna asunción, a priori, sobre la relación funcional entre las características y el número de personas [83].

A continuación se explican cada una de las etapas del sistema de regresión:

1. **Distribución de probabilidad a priori**

   Un proceso gaussiano se define como una colección finita de variables aleatorias (proceso estocástico) que tienen una distribución de probabilidad gaussiana conjunta. Una distribución normal o gaussiana se define completamente por su vector de media y su matriz de covarianza. De manera análoga, los procesos gaussianos se definen mediante su función de media $m(x)$ y su función de covarianza $k(x_n, x_m)$, es decir, un proceso gaussiano define una distribución de probabilidad normal o gaussiana sobre un espacio de funciones [80]. En los problemas de regresión, se tiene normalmente un set de entrenamiento de N muestras, donde n se refiere al número de segmento, compuestas por el vector de características $X = \{x_n\}$ y el *ground truth* $f = \{f_n\}$. Estos términos son los mismos que los de las ecuaciones 3.12 y 3.20 respectivamente.

   La distribución de probabilidad del set de entrenamiento se puede expresar de la siguiente manera:

$$f|X \sim \mathcal{N}(0, K) \tag{3.22}$$

   Donde la matriz de covarianza, $K \epsilon \mathbb{R}^{N \times N}$, se obtiene a partir de la función de covarianza, $K_{nm} = k(x_n, x_m)$ y la función de media es típicamente cero, $m(x) = 0$. La función de covarianza expresa la covarianza de la salida en función de la entrada. Una función de covarianza típica es la función exponencial cuadrática (*squared exponential*) :

$$k_{SE}(x_n, x_m) = \sigma_{SE}^2 exp\left(-\frac{1}{2l^2}\left|x_n - x_m\right|^2\right) \tag{3.23}$$

   En la ecuación anterior, cuanto más cera estén las entradas x_n y x_m, más correladas serán las salidas. El hiperparámetro, l, es una escala de longitud característica que representa la distancia esperada que se debería mover en el espacio de entrada para producir un cambio significativo en el espacio de salida [80].

   Hasta el momento, solo se ha definido una distribución sobre un espacio de funciones

utilizando un proceso gaussiano, lo que constituye la distribución de probabilidad a priori para la inferencia Bayesiana. Esta, no depende de los datos de entrenamiento, sino que simplemente especifica propiedades de las funciones.

2. **Distribución de probabilidad a posteriori**

   A continuación, se presentan las reglas para actualizar la distribución a priori utilizando los datos de entrenamiento y obtener la distribución de probabilidad a posteriori. La distribución a posteriori permitirá hacer predicciones en casos de test no observados [80].

   Dadas N^* muestras de entrada de test, $X^* = \{x_n^*\}$, se desea predecir los valores de la función $f^* = \{f_n^*\}$. En este caso X^* denota al vector de características para los n segmentos de test y $f*$, el vector que contiene el número de personas dentro de cada uno de los segmentos. La matriz de covarianza $N \times N^*$ de entrenamiento-test, se denotaría como K^*, con $K_{nm}^* = k(x_n, x_m^*)$, y de manera similar la matriz de covarianza $N^* \times N^*$ de test sería K^{**}, con $K_{nm}^{**} = k(x_n^*, x_m^*)$.

   Como todas las variables de un proceso gaussiano tienen una distribución normal, los datos de test seguirán también una distribución normal:

$$f^*|X^* \sim \mathcal{N}(0, K^{**}) \tag{3.24}$$

   Por tanto, los datos de entrenamiento y test tendrán una distribución conjunta gaussiana que se puede expresar de la siguiente manera:

$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K & K^* \\ K^{*T} & K^{**} \end{bmatrix}\right) \tag{3.25}$$

   Dado que los valores del set de entrenamiento f son conocidos, es interesante expresar la distribución condicional de f^* dado f, también denominada distribución de probabilidad a posteriori:

$$f^*|f, X, X^{**} \sim \mathcal{N}(\mu, \Sigma) \tag{3.26}$$

$$\mu = K^{*T} K^{-1} f \tag{3.27}$$

$$\Sigma = K^{**} - K^{*T} K^{-1} K^{*} \tag{3.28}$$

Para obtener la distribución condicional de una distribución gaussiana conjunta se ha empleado la fórmula:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim N\left(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix}\right) \Rightarrow x|y \sim \mathcal{N}(a + CB^{-1}(y-b), A - CB^{-1}C^T) \tag{3.29}$$

Las ecuaciones 3.26, 3.27 y 3.28 constituyen las ecuaciones centrales de las predicciones del proceso gaussiano y proporcionan, no solo, los valores de la estimación, μ, sino también, la matriz de covarianza para las salidas, Σ, cuya diagonal principal constituye la varianza de las estimaciones.

$$\sigma_n^2 = \Sigma_{nn} \tag{3.30}$$

Por poner un ejemplo, si se establece un intervalo de confianza del 95 %, las estimaciones para el set de datos de test serían:

$$\mu_n \pm 1{,}96\sigma_n \tag{3.31}$$

.

La predicción y la varianza para toda la escena se puede obtener calculando la suma de N^* variables aleatorias:

$$\mu_{hol} = \sum_{n=1}^{N^*} \mu_n \tag{3.32}$$

$$\sigma_{hol}^2 = \sum_{n=1}^{N^*} \sigma_n^2 \tag{3.33}$$

En el sistema implementado se utilizan dos funciones de covarianza distintas, con el fin de capturar las tendencias de los datos en un rango incluso mayor al de los datos de entrenamiento. En primer lugar, se utiliza la función de covarianza exponencial cuadrática, definida ya en la ecuación 3.23, que captura la noción intuitiva de que entradas similares deben producir salidas similares. Con el objetivo de capturar tendencias en un rango mayor que el de los datos de entrenamiento, se utiliza la función de covarianza de producto de puntos:

$$k_{DP}(x_n, x_m) = \sigma_{DP}^2(1 + x_n^T x_m) \tag{3.34}$$

La combinación de ambas funciones permite que el sistema de regresión mantenga la no linealidad dentro del rango de los datos de entrenamiento y, fuera de dicho rango, se extrapole de manera predominantemente lineal.

Por último, se debe tener en cuenta el ruido en el set de datos de entrenamiento, ya que es frecuente que las observaciones presenten ruido. Habitualmente se asume que se trata de un ruido aditivo independiente e idénticamente distribuido gaussiano. El efecto de este ruido se traduce en una covarianza extra con una magnitud igual a la varianza del ruido:

$$k_{GN}(x_n, x_m) = \sigma_{GN}^2 \delta(n, m) \tag{3.35}$$

Donde δ representa la función delta de Kronecker que contribuye solamente a las diagonales de K y K^{**}. Por tanto la función de covarianza final sería:

$$k(x_n, x_m) = k_{SE}(x_n, x_m) + k_{DP}(x_n, x_m) + k_{GN}(x_n, x_m) \tag{3.36}$$

$$= \sigma_{SE}^2 exp\left(-\frac{1}{2l^2}\left|x_n - x_m\right|^2\right) + \sigma_{DP}^2(1 + x_n^T x_m) + \sigma_{GN}^2 \delta(n, m) \tag{3.37}$$

Como se puede observar en la expresión anterior, la función de covarianza está definida en función de unos hiperparámetros, por lo que para conocer completamente la distribución de probabilidad a priori, es necesario elegir el valor de estos hiperparámetros. A este proceso se le denomina entrenamiento.

3. **Entrenamiento del sistema**

   Como ya se ha adelantado ya, la distribución de probabilidad a priori se define mediante las funciones de media y covarianza que están parametrizadas por unos hiperparámetros [80]. El valor de dichos hiperparámetros se debe inferir durante el entrenamiento de acuerdo con los datos de entrenamiento. Como en este caso concreto, la función de media elegida es $m(x) = 0$, solo será necesario aprender los hiperparámetros de la función de covarianza, o sea $\theta = \{\sigma_{SE}, \sigma_{DP}, \sigma_{GN}, l\}$.

   Para hacer inferencias sobre los hiperparámetros, se calcula la probabilidad de los datos, dados los hiperparámetros $p(f|X)$ (*likelihood*), lo cual es relativamente sencillo ya que se ha asumido que los datos siguen una distribución gaussiana:

$$log\, p(f|X) = -\frac{1}{2} f^T K^{-1} f - \frac{1}{2} log\, |K| - \frac{N}{2} log\, 2\pi \tag{3.38}$$

   El término anterior se denomina *log likelihood* y se maximiza utilizando un algoritmo de optimización como por ejemplo el algoritmo de gradientes conjugados siempre que $m(x)$ y $k(x_n, x_m)$ sean diferenciable con respecto a cada uno de sus respectivos hiperparámetros (en este caso, solo $k(x_n, x_m)$, ya que $m(x) = 0$).

4. **Test**

   Una vez optimizados los hiperparámetros, se realizan las predicciones a partir de los datos de test utilizando las ecuaciones 3.26, 3.27 y 3.28.

Para realizar la regresión, tal y como se ha explicado hasta ahora, se ha utilizado la herramienta **GPML** (*Gaussian Processes for Machine Learning*) [78] disponible para Matlab, que proporciona un amplio rango de funcionalidades para la inferencia (entrenamiento, obtención de hiperparámetros) y predicción (estimación, test) de procesos gaussianos. La herramienta proporciona una librería de funciones de media y covarianza simples y ofrece la posibilidad de combinar más de una en funciones más complejas. En este caso se han utilizado las funciones de media y covarianza definidas arriba y se ha realizado el entrenamiento para cada una de las pruebas que se describirán en el siguiente capítulo a partir de las características extraídas y el *ground truth* calculado. Una vez aprendidos los valores de los hiperparámetros en cada prueba, se han utilizado como parámetros de

entrada del algoritmo de regresión para realizar las estimaciones en la fase final de test utilizando también la herramienta **GPML**.

4

Evaluación

4.1. Introducción

El objetivo principal de este capítulo es evaluar el algoritmo de estimación de densidad mediante la realización de todas las pruebas necesarias para comparar los resultados de la estimación utilizando los dos segmentadores descritos en el capítulo anterior: segmentador frente-fondo (ver sección 3.2.1) y segmentador persona-fondo (ver sección 3.2.2). De manera general, la ventaja principal del segmentador persona-fondo es que segmenta la imagen basándose en la detección de personas, frente al segmentador frente-fondo, que busca separar el frente de la imagen, basándose en la detección de movimiento de los objetos (no solo personas) de la escena. No obstante, dada la complejidad de la detección de personas en una escena cualquiera, los resultados de la segmentación con el segmentador frente-fondo utilizado en este caso, son normalmente más precisos. La motivación para llevar a cabo la comparación de ambos segmentadores se basa principalmente en dos consideraciones: por un lado, las características concretas de la escena en la que se realice la segmentación, podrían influir de manera significativa e indistintamente en los resultados obtenidos por uno u otro segmentador; por otro lado, se desea comprobar si una segmentación muy ajustada al contorno de las personas como la que se obtiene con

el segmentador frente-fondo implica necesariamente que las estimaciones obtenidas por el algoritmo de estimación de densidad de personas sean mejores. Teniendo en cuenta ambas consideraciones, en la sección de resultados de este capítulo, se hará especial hincapié en analizar los resultados obtenidos sobre varias escenas (varias bases de datos) para poder extraer conclusiones sobre las diferencias entre ambos algoritmos aplicados a la estimación de densidad de personas.

Antes de presentar los resultados, en la sección 4.2 se realizará un resumen de las bases de datos empleadas para llevar a cabo la evaluación de los resultados, así como de los datos disponibles para cada una, como pueden ser la anotación para el cálculo del *ground truth* o los parámetros de calibración. En la sección 4.3, se presentarán las métricas utilizadas para medir o cuantificar los resultados de acuerdo a la precisión de la estimación. Finalmente, en la sección 4.4, se describen cada una de las pruebas realizadas y se analizan con el fin de extraer conclusiones acerca de los resultados obtenidos. A modo de resumen, en la sección 4.5 se presentan las principales conclusiones extraídas a partir de los resultados.

4.2. Base de datos

En este contexto, una base de datos se refiere a un conjunto de imágenes, procedentes de una secuencia capturada en una escena real, que se utilizan como entrada del algoritmo de estimación de densidad implementado en este proyecto, tanto para el entrenamiento como para el test. La comparación de las estimaciones de salida del algoritmo en la fase de test, con el número de personas de cada una de las imágenes (proporcionada por el *ground truth* de la imagen extraído a partir de las anotaciones), permitirá en este caso evaluar los resultados de la estimación. Los *datasets* utilizados para las pruebas realizadas han sido creados para evaluar algoritmos de procesamiento digital de vídeo y varios de ellos se han utilizado para evaluar algunos de los algoritmos de estimación de densidad del Estado del Arte.

Las bases de datos que se han utilizado para la evaluación del algoritmo son las utilizadas por los autores del algoritmo de estimación implementado [83]: **UCSD** *pedestrian database* y **SAIVT-QUT** *crowd counting database*, junto con la base de datos **TUD**. A continuación, se presenta una breve descripción de cada una de ellas.

- **UCSD** *pedestrian database* [11]: Esta base de datos está disponible[1] para realizar experimentos sobre algoritmos de visión artificial. Contiene una secuencia de 2000 imágenes anotadas de peatones que se mueven en dos direcciones. El vídeo está submuestreado a 238x158 píxeles y 10 fps y las imágenes están en escala de grises. Junto con el *dataset* se proporcionan la máscara para la **ROI** y ficheros con las anotaciones de los centros de las personas para cada *frame* y el mapa de perspectiva descrito en la sección 3.4 del capítulo anterior. En la figura 4.1 se puede ver un ejemplo visual de una imagen de esta base de datos. Al no disponer en este caso de las parámetros de calibración, se ha utilizado el mapa de perspectiva proporcionado con este *dataset* para la normalización de las características extraídas, por la misma razón, para calcular el *ground truth*, tal y como se explica en la sección 3.5.1 del capítulo anterior, no se han utilizado los cilindros proyectados de 3D a 2D, sino que se ha definido una caja alrededor de cada persona de un tamaño fijo, similar al tamaño de las personas dentro de la escena.

- **SAIVT-QUT** *crowd counting database* [83]: Dentro de **SAIVT-QUT**[2] se incluyen varios *datasets* públicos como son **PETS2006** y **PETS2009** además de otro denominado **QUT**. Concretamente, en esta base de datos se incluyen *frames*, anotaciones de personas, ROI y parámetros de calibración de Tsai para las secuencias **PETS2009-S1-L1-View001 13-57** y **13-59**, **PET2009-S1-L1-View002 13-57** y **13-59**, **PETS2006-S1-T1-View003**, **PETS2006-S1-T1-View004**, **QUT-cameraA**, **QUT-cameraB** y **QUT-cameraC**. En el caso de los *datasests* **PETS 2006** y **PETS 2009** no se incluye el vídeo, sino solo el resto de datos (anotaciones, ROI, calibración y anotaciones), por lo que las secuencias se han obtenido de sus respectivas páginas web[3][4]. De esta base de datos se han utilizado todas las secuencias excepto **PET2009-S1-L1-View002 13-57** y **13-59** y **PETS2006-S1-T1-View004**, por no aportar ninguna dificultad nueva para los segmentadores aparte de las que se tiene en el resto de secuencias. En la figura 4.1 se puede ver un ejemplo visual de un *frame* de cada una de las secuencias empleadas. Dado que se dispone de los parámetros de calibración de la cámara, se ha calculado el mapa

[1] **UCSD**: http://www.svcl.ucsd.edu/projects/peoplecnt/
[2] **SAIVT-QUT**: https://wiki.qut.edu.au/display/saivt/SAIVT-QUT+Crowd+Counting+Database
[3] **PETS2006**: http://www.cvg.reading.ac.uk/PETS2006/data.html
[4] **PETS2009**: http://www.cvg.reading.ac.uk/PETS2009/a.html

Base de datos	**Nº frames**	**Anotadas**	**Tamaño**	**N máximo de personas**	**Calibración**
UCSD	2000	2000	238x158	45	No
PETS2009 view 1 (13-57, 13-59)	220+240	46	568x576	32	Si
PETS2006 view 3	3000	120	720x576	5	Si
QUT camera A	10400	50	704x576	8	Si
QUT camera B	5300	50	704x576	23	Si
QUT camera C	5300	50	704x576	10	Si
TUD campus	71	71	640x480	7	No
TUD crossing	201	201	640x480	11	No

Cuadro 4.1: Secuencias utilizadas para evaluar el algoritmo.

de densidades para realizar la normalización de las características tal y como se ha descrito en la sección 3.4 del capítulo anterior.

- **TUD**[5]: Dentro de esta base de datos se han utilizado las secuencias *campus* y *crossing*. Estas secuencias son relativamente más sencillas que las anteriores ya que la densidad de personas es menor y la cámara está más cerca. Dado que no se dispone de parámetros de calibración, ni tampoco de mapa de perspectiva y que además, el ángulo de la cámara con respecto al plano de tierra en ambas secuencias es casi cero (por lo que el tamaño de las personas no varía a penas), no se han normalizado las características extraídas.

En la tabla 4.1 se muestra un resumen de las secuencias empleadas en este proyecto y la información más relevante de cada una.

4.3. Métrica

Con el objetivo de cuantificar el error de las estimaciones, para todas las pruebas realizadas se han utilizado las métricas empleadas en [83]:

1. **MAE** (*Mean Absolute Error*): El error absoluto medio permitirá medir la magnitud media del error de las estimaciones de densidad para cada imagen, es decir la

[5] **TUD**: https://www.d2.mpi-inf.mpg.de/andriluka_cvpr08#name:cvpr08_data

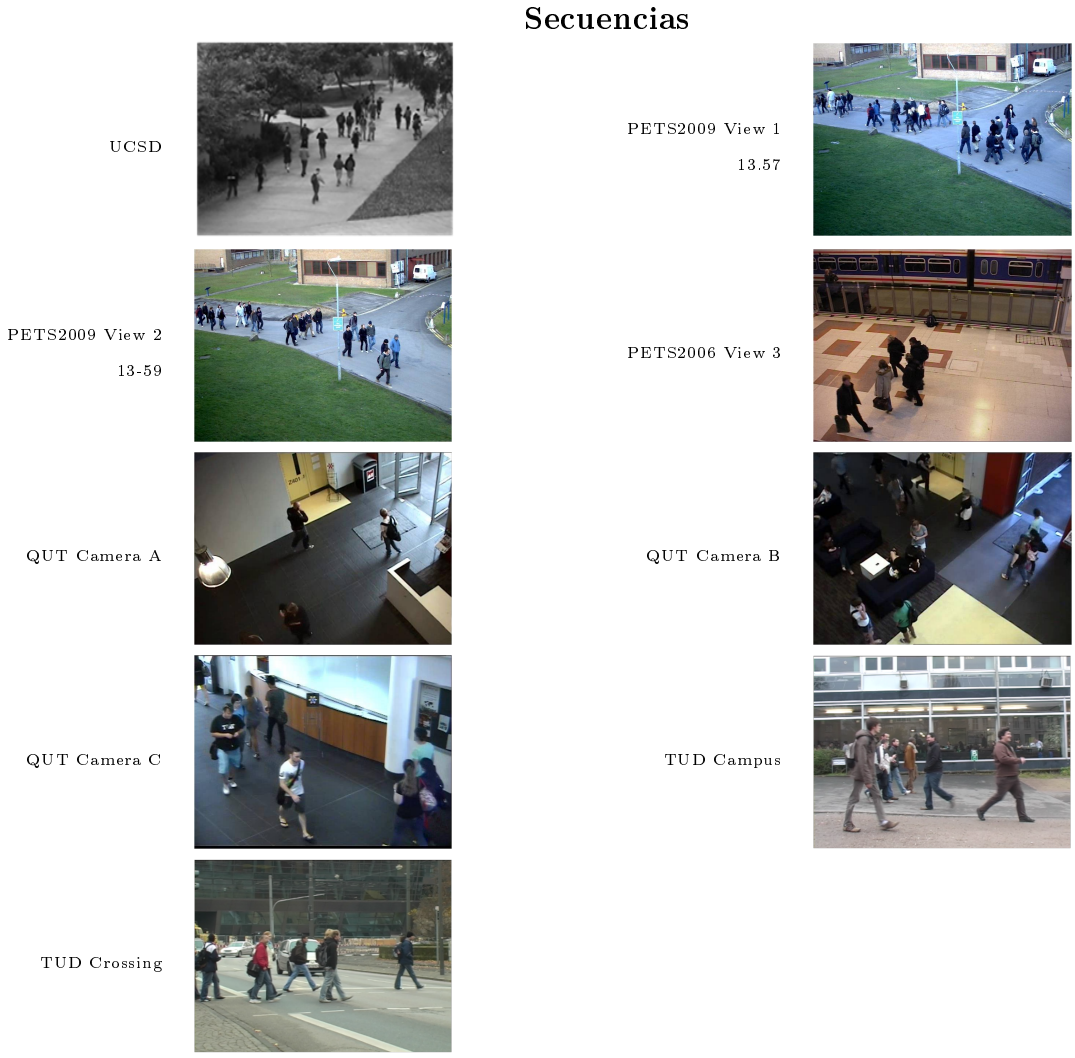

Figura 4.1: Ejemplo visual de los *datasets*.

precisión de la estimación. El **MAE** ofrece una puntuación lineal, lo que significa que todas las diferencias entre el valor estimado y el número real de personas serán penalizadas por igual en la media. El cálculo del **MAE** se realiza utilizando la siguiente expresión:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{4.1}$$

donde f_i y y_i son las estimaciones y los valores reales observados respectivamente.

1. **MSE** (*Mean Squared Error*): Dado que los errores se elevan al cuadrado antes de realizar la media, el error cuadrático medio da más peso a los errores más grandes. Es decir, para un mismo valor de **MAE**, un **MSE** mayor significa que se han cometido errores de estimación individuales mayores, aunque la media sea la misma. El cálculo del **MSE** se realiza utilizando la siguiente expresión:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \tag{4.2}$$

4.4. Resultados

En esta sección se presentarán todas las pruebas realizadas para testear el algoritmo de estimación de densidad con ambos segmentadores. Todas las evaluaciones, se han realizado utilizado el segmentador persona-fondo y el segmentador frente-fondo. Como se ha comentado en la sección 3.2.2 del capítulo 3, se han utilizado dos versiones del segmentador persona-fondo, **DEBP** y **DEBP-P**. Además, para ejecutar el algoritmo, se debe elegir un umbral que se aplica sobre la confianza de las detecciones y determina un nivel de aceptación (con un umbral más bajo se tenderá a clasificar una mayor cantidad de píxeles como persona que con un umbral mayor). Para determinar el valor más óptimo del umbral en cada caso, se ha realizado la segmentación con varios umbrales, concretamente se han elegido los valores 0.75:0.05:0.9 (en notación de Matlab), es decir, umbrales muestreados desde 0.75 a 0.9 en pasos de 0.05, ya que empíricamente se ha observado que es el rango de umbrales óptimo. Por tanto, para todas las secuencias antes mencionadas se ha realizado la segmentación persona-fondo para todos los umbrales y para las dos versiones del algoritmo, con post-procesado y sin post-procesado y la segmentación frente-fondo.

Ambos segmentadores presentan ciertas ventajas y desventajas que han condicionado las pruebas realizadas. En primer lugar, el algoritmo se segmentación frente-fondo requiere de un tiempo de aprendizaje al comienzo de las secuencias, que afecta a los resultados de la segmentación de los primeros *frames*, pero no por igual a todas las secuencias. Por este motivo, se ha intentado evitar utilizar los primeros *frames* de cada secuencia tanto en test como en entrenamiento. Por otro lado, el segmentador persona-fondo no es capaz de obtener resultados aceptables cuando el tamaño de las personas es muy pequeño, por lo que el *dataset* **UCSD** cuyo tamaño original era 238x158 píxeles se ha tenido que redimensionar a 714x474 píxeles.

Tal y como se ha comentado en la sección 4.2 se han evaluado los resultados para distintas secuencias. En las secciones 4.4.1, 4.4.2, 4.4.3, 4.4.4 y 4.4.5 se presentan los resultados de las pruebas y las conclusiones a las que se ha llegado con cada una. Finalmente, en la sección 4.5 se presentan las conclusiones generales extraídas al comparar la utilización de ambos tipos de segmentación para la estimación de densidad de personas.

4.4.1. Resultados obtenidos para la secuencia UCSD

Como ya se ha comentado con anterioridad, el objetivo primordial del proyecto es la comparar la utilización de ambos segmentadores en el algoritmo de estimación de densidad. Sin embargo, antes de entrar a analizar las diferencias entre uno y otro algoritmo, se desea comprobar que el algoritmo de estimación de densidad implementado funciona correctamente. Para ello, se han realizado los mismos tests que proponen los autores del algoritmo original [83] y se presentan los resultados obtenidos junto a los resultados del algoritmo original y otros algoritmos similares del Estado del Arte.

Para evaluar la precisión del sistema se ha utilizado en primer lugar, la secuencia **UCSD**, reservando los *frames* del 601 al 1400 (de 2000) para entrenamiento y los restante 1200 *frames* para test. Concretamente, se utilizan tres *subsets* de entrenamiento diferentes: 610:80:630, 640:80:1360 y 670:80:1390 en notación de Matlab, siendo cada uno de 10 imágenes. La utilización de varios sets de entrenamiento para todas las secuencias empleadas, permitirá obtener resultados más representativos del funcionamiento del sistema. Estos sets de entrenamiento y test son los mismos que se han utilizado para evaluar el algoritmo original. En la tabla 4.2 se presentan los resultados obtenidos para el segmentador

frente-fondo y todos los umbrales y versiones del segmentador persona-fondo. En negro, se marcan los mejores resultados y la tabla está ordenada por el valor medio del **MAE** para los tres sets de entrenamiento. Como se puede apreciar, los mejores resultados se obtienen con el segmentador frente-fondo para todos los sets de entrenamiento. Aunque las diferencias en el error medio no son tan pronunciadas, en el error cuadrático medio si lo son, lo que significa que en algunos *frames* el error es significativamente mayor. En las figuras 4.2 y 4.3 se representan gráficamente el *ground truth* y las estimaciones realizadas para todas las imágenes de test en los tres sets de entrenamiento, para el segmentador frente-fondo y el segmentador persona-fondo con umbral 0.80 sin post-procesado (con el que se obtiene el mejor resultado) respectivamente. Alrededor de las estimaciones se presenta la varianza de las estimaciones obtenidas del algoritmo de regresión aplicando un intervalo de confianza del 95 %. En la ecuación 3.31 del capítulo anterior, se muestra cómo se aplica el intervalo de confianza a la varianza que se obtiene del algoritmo de regresión.

A continuación, en la tabla 4.3, se presentan los resultados obtenidos por los algoritmos del Estado del Arte [11, 47, 50, 83] junto a los dos mejores resultados del algoritmo implementado. Al comparar los resultados obtenidos con los de otros algoritmos del Estado del Arte se puede comprobar que el algoritmo se ha implementado correctamente, ya que se obtienen errores similares.

Finalmente, se han realizado pruebas con 4 nuevos sets de entrenamiento: uno al comienzo de la secuencia (100:5:195), uno en medio (500:5:595), uno al final (1900:5:1995) y uno distribuido a lo largo de toda la secuencia (100:100:2000). Esta vez, se ha aumentado el número de imágenes de entrenamiento de 10 a 20 para comprobar si mejoran los resultados. El set de test está fuera del rango de entrenamiento para los 3 primeros sets de entrenamiento y para el último, solo se han excluido del test los *frames* que forman parte del set de entrenamiento. El primer set de entrenamiento, comienza con el *frame* 100 ya que como se ha comentado arriba, el algoritmo de segmentación frente-fondo necesita un tiempo de inicialización. Tal y como se ha explicado hasta ahora, se elegirán los sets de entrenamiento y test para las pruebas sobre las secuencias restantes, aunque de acuerdo con el número de *frames* disponibles para cada una. En la tabla 4.4 se muestran los errores obtenidos para las estimaciones.

De los resultados obtenidos se pueden extraer varias conclusiones:

	610:80:1330		**640:80:1360**		**670:80:1390**		**valor medio**	
Seg	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
f-f[a]	**1.18**	**2.12**	**1.12**	**1.88**	**1.35**	**2.62**	**1.22**	**2.21**
p-f[b] **0.80**	2.99	13.46	3.28	15.29	3.54	18.03	3.27	15.60
p-f 0.80 post	3.27	15.56	2.90	13.03	3.92	21.78	3.36	16.79
p-f 0.70	3.01	13.18	3.60	17.79	3.71	19.15	3.44	16.70
p-f 0.75	2.73	11.14	3.58	17.65	4.35	25.45	3.55	18.08
p-f 0.75 post	3.59	18.50	3.14	14.86	3.95	21.80	3.56	18.39
p-f 0.85	3.74	23.30	3.83	25.19	3.16	15.40	3.58	21.30
p-f 0.85 post	3.74	20.76	3.52	20.22	3.55	19.21	3.61	20.06
p-f 0.70 post	3.32	16.28	4.46	28.43	4.70	31.02	4.16	25.24
p-f 0.90 post	6.62	68.43	6.10	61.06	6.46	65.87	6.39	65.12
p-f 0.90	7.14	73.49	7.18	76.23	5.44	47.92	6.58	65.88

[a] **f-f: segmentador frente-fondo**

[b] **p-f: segmentador persona-fondo**

Cuadro 4.2: Errores de estimación del algoritmo implementado sobre la base de datos **UCSD** con sets de entrenamiento 610:80:630, 640:80:1360 y 670:80:1390.

Sistema	**Training set**	**MAE**	**MSE**
Kong lineal [47]	all	1.92	5.60
Kong neural network (5 runs) [47]	all	2.47±0.41	9.53±3.01
Chan [11]	all	1.95	6.06
Lempitsky [50]	605:5:1400	1.70	-
Lempitsky [50]	640:80:1360	2.02	-
Algoritmo original [83]	610:80:1330	1.79	4.95
Algoritmo original [83]	640:80:1360	1.33	2.91
Algoritmo original [83]	670:80:1390	1.57	3.94
implementado seg frente-fondo	610:80:1330	1.18	2.12
implementado seg frente-fondo	640:80:1360	1.12	1.88
implementado seg frente-fondo	670:80:1390	1.35	2.62
implementado seg persona-fondo	610:80:1330	2.99	13.46
implementado seg persona-fondo	640:80:1360	3.28	15.29
implementado seg persona-fondo	670:80:1390	3.54	18.03

Cuadro 4.3: Errores de estimación de los algoritmos del Estado del Arte sobre la base de datos **UCSD** con sets de entrenamiento 610:80:630, 640:80:1360 y 670:80:1390.

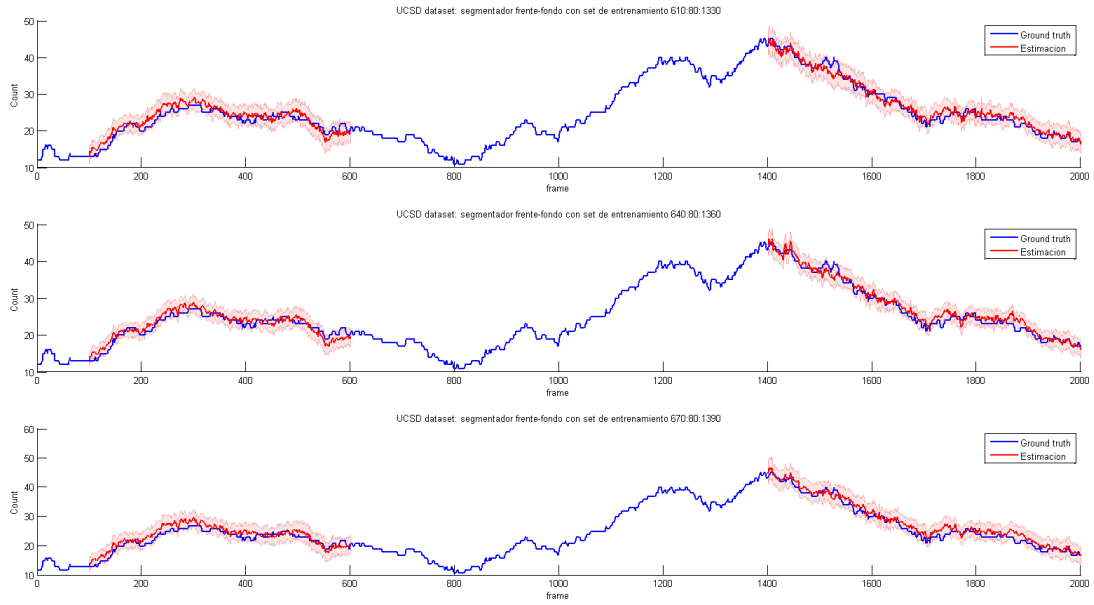

Figura 4.2: Representación gráfica de las estimaciones y su varianza obtenidas con el segmentador frente-fondo junto al *ground truth* para cada *frame* de test, dados los sets de entrenamiento: 610:80:630, 640:80:1360 y 670:80:1390.

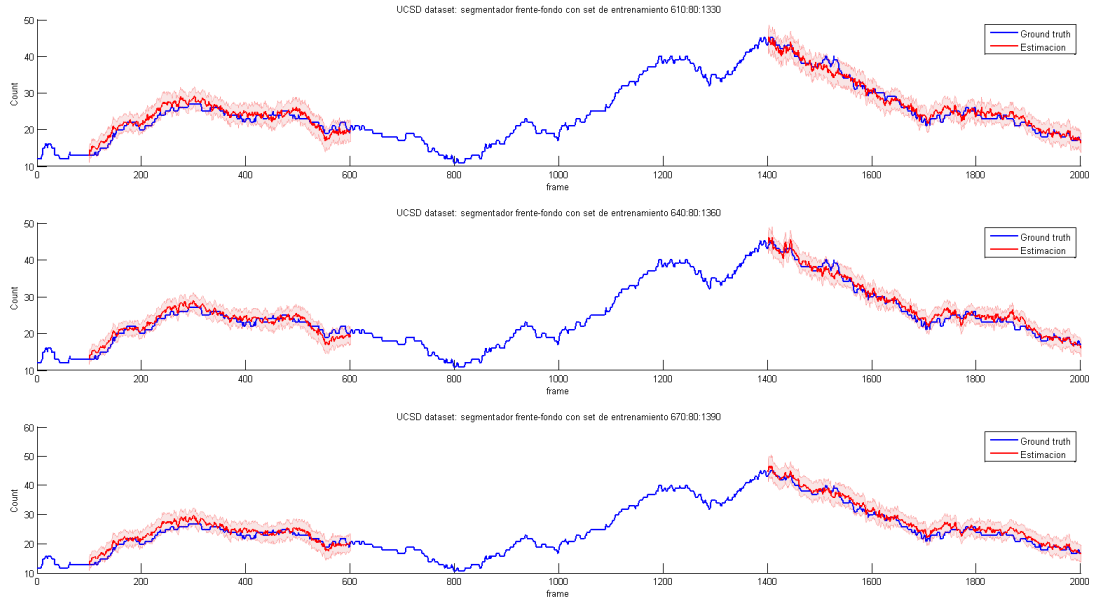

Figura 4.3: Representación gráfica de las estimaciones y su varianza obtenidas con el segmentador persona-fondo con umbral 0.80 junto al *ground truth* para cada *frame* de test, dados los sets de entrenamiento: 610:80:630, 640:80:1360 y 670:80:1390.

	100:5:195		**500:5:595**		**1900:5:1995**		**100:100:2000**		**valor medio**	
Seg	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
f-f[a]	3.84	**19.39**	**1.32**	**2.77**	**1.31**	**2.78**	**1.26**	**2.40**	**1.93**	**6.83**
p-f[b] **0.75:**	3.79	23.88	3.44	20.19	3.96	26.39	2.94	13.03	3.53	20.87
p-f 0.70:	4.48	35.70	3.46	21.10	4.34	31.94	2.94	13.47	3.80	25.55
p-f 0.80:	**3.62**	21.55	3.88	26.06	4.84	36.62	3.14	14.83	3.87	24.77
p-f 0.75 post:	5.20	46.87	4.43	33.92	4.83	39.41	2.94	13.41	4.35	33.40
p-f 0.80 post:	4.81	38.84	4.45	33.51	5.28	44.26	2.95	13.40	4.37	32.50
p-f 0.70 post:	6.49	75.37	4.67	39.07	5.63	54.23	3.13	15.15	4.98	45.96
p-f 0.85:	4.64	33.15	6.30	59.33	6.60	61.51	3.83	23.43	5.34	44.36
p-f 0.85 post:	6.22	58.18	5.85	53.85	6.92	68.20	3.74	22.00	5.68	50.56
p-f 0.90:	5.79	53.02	8.46	101.68	7.61	82.54	7.29	77.90	7.29	78.79
p-f 0.90 post:	10.09	133.23	9.49	121.02	11.22	155.64	6.61	67.86	9.35	119.44

[a] **f-f: segmentador frente-fondo**

[b] **p-f: segmentador persona-fondo**

Cuadro 4.4: Errores de estimación del algoritmo implementado sobre la base de datos **UCSD** con sets de entrenamiento 100:5:195, 500:5:595 1900:5:1995 100:100:2000.

1. En primer lugar, si se comparan los resultados de las pruebas incluidas en las tablas 4.4 y 4.2 se puede ver que la diferencia de los errores obtenidos en ambas pruebas para un mismo segmentador no son significativamente diferentes, por lo que se puede considerar que 10 imágenes de entrenamiento son suficientes (en adelante, se utilizarán solo 10).

2. Como se puede ver en la tabla, los resultados de la estimación para el primer set de entrenamiento con el segmentador frente-fondo, son peores que los obtenidos en el resto de sets de entrenamiento con este mismo segmentador y peores también, que los obtenidos con algunas versiones del segmentador persona-fondo, ya que a pesar de que se han descartado los primeros 100 *frames* de la secuencia, aún los modelos de fondo no se han aprendido completamente. Para el resto de sets de entrenamiento, los errores con el segmentador frente-fondo siguen siendo inferiores que los errores con el segmentador persona-fondo. En la figura 4.4 se puede ver un ejemplo de cómo evoluciona la segmentación obtenida del segmentador frente-fondo. Se aprecia claramente como en el *frame* número 30, el sistema aún no es capaz de detectar a las personas. En el *frame* 100 la segmentación es considerablemente mejor, aunque aún hay muchos píxeles de personas no detectados, mientras que en

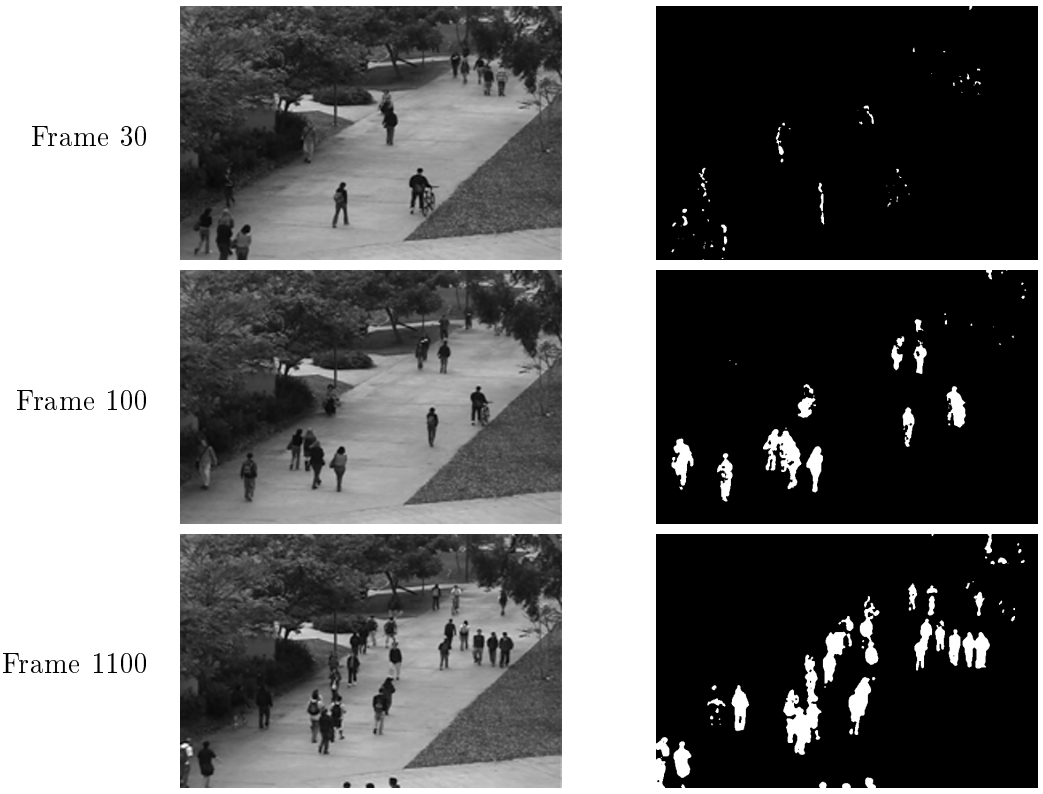

Figura 4.4: Evolución de la máscara de segmentación con el segmentador frente-fondo en secuencia **UCSD**.

el *frame* 1100 se rellena mejor el interior de los objetos detectados.

3. Como se ha explicado en la sección 4.2, los *frames* de la secuencia **UCSD** se han redimensionado a 3 veces el tamaño original porque que el algoritmo de segmentación persona-fondo, basado en la detección de las personas, no podía detectar a las personas siendo tan pequeñas. Aún así, esta secuencia, resulta especialmente complicada para este segmentador, ya que, por un lado, las personas siguen siendo bastante pequeñas y al aumentar la resolución se ha perdido calidad en las imágenes y por otro lado, en algunas partes de la secuencia la densidad de personas es bastante elevada (hasta 45 personas), habiendo además grandes oclusiones en grupos de personas, lo cual afecta de manera importante a los resultados.

4. En la figura 4.5 se puede ver un ejemplo de la máscara obtenida por ambos segmentadores para esta secuencia. En el caso del segmentador persona-fondo, se muestran ejemplos de las máscaras extraídas para las versiones que mejor y peor resultado

han proporcionado en la estimación, que son en este caso el segmentador con umbral 0.75 sin post-procesado y el de umbral 0.90 con post-procesado. Como se puede ver en la figura, cuando el umbral es 0.75 se detectan más personas que las que hay en el *frame*, mientras que con el umbral en 0.90 hay personas que no se detectan. A la luz de los resultados de la estimación para uno y otro umbral, se puede concluir que es preferible tener falsos positivos que falsos negativos, ya que por un lado, cuando los segmentos son más pequeños que el tamaño de las personas, el algoritmo de regresión puede aprender que no son personas basándose en las características de área y perímetro extraídas de los segmentos, y por otro lado, cuando los segmentos son tan o más grandes que las personas, se puede aprender que no son personas utilizando los histogramas de orientación de gradientes. En cambio, el hecho de que no se detecten las personas que están en la escena, no se puede corregir de ninguna manera, independientemente del proceso de entrenamiento. De esta forma, se pueden obtener errores tan pequeños como por ejemplo el error medio para el set de entrenamiento distribuido 100:100:2000 con umbral 0.75 que es menor que 3 (ver tabla 4.4). Además, se puede observar en la tabla 4.4 que tanto el **MAE** como el **MSE**, cuando se utiliza el segmentador persona-fondo son significativamente menores para la secuencia de entrenamiento distribuida (100:100:2000) que para el resto de secuencias de entrenamiento, lo que sugiere que al utilizar un set de entrenamiento que contiene *frames* con mayor variabilidad, se puede aprender mucho mejor a diferenciar los segmentos que corresponden a personas de los que no. Aún así, los resultados evidencian que el segmentador frente-fondo, sigue siendo más apropiado para secuencias de este tipo, no tanto por el hecho de proporcionar una máscara mucho más ajustada al contorno de las personas, como por proporcionar una máscara con muchos menos falsos positivos y falsos negativos.

5. Como ya se ha comentado el **MSE** obtenido en todas las pruebas al utilizar el segmentador persona-fondo es bastante grande, lo que significa que hay *frames* en los que el error de la estimación es elevado y que por tanto tienen mucho más peso en el error cuadrático medio que en el error absoluto medio. Para ver cuándo el error es mayor, se ha analizado la segmentación obtenida utilizando un umbral de 0.75, para la secuencia de entrenamiento 100:100:2000, con la que se que se obtienen el

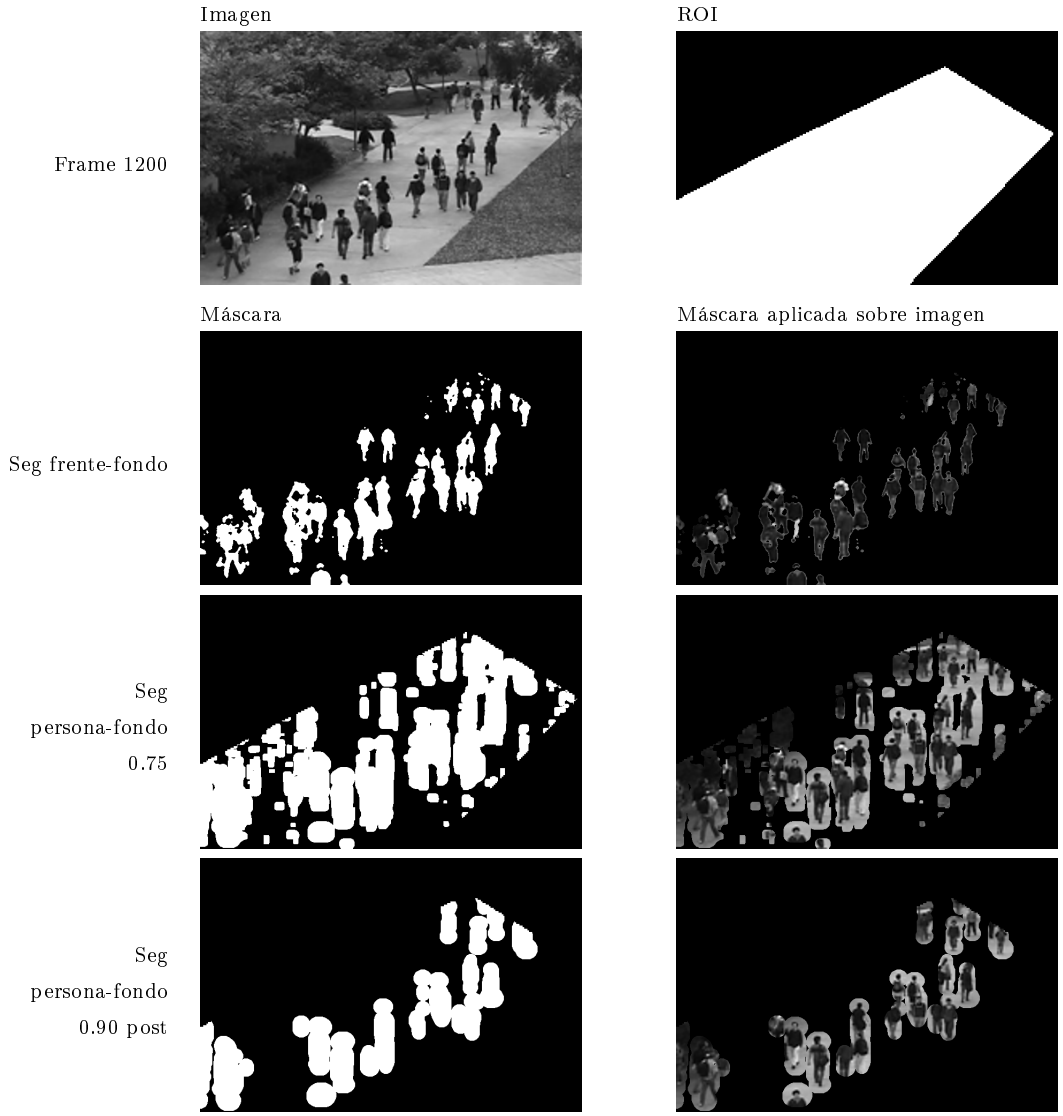

Figura 4.5: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **UCSD.**

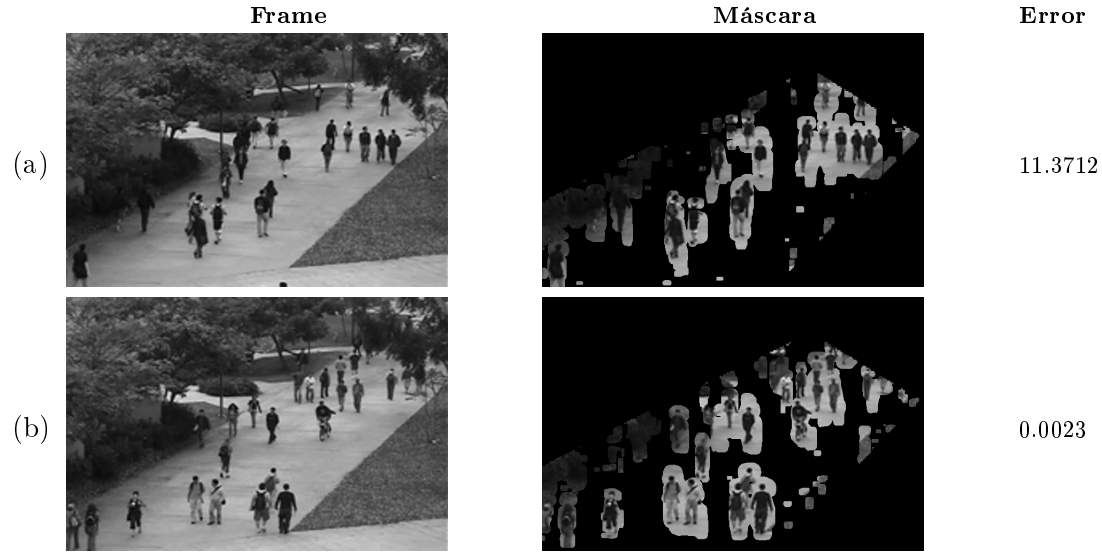

Figura 4.6: Comparación máscaras del segmentador persona-fondo con umbral 0.75 con secuencia **UCSD**. (a) Máscara con la que se obtiene el menor error de estimación. (b) Máscara con la que se obtiene el mayor error de estimación.

mayor y el menor error de estimación. En la figura 4.6 (a) se muestran el *frame* y la máscara para los que se obtiene el mayor error y en la figura 4.6 (b) se muestran el *frame* y la máscara para los que se obtiene el menor error. Como se puede observar, ambas segmentaciones son muy parecidas, lo único que las diferencia es que en el *frame* original hay más densidad en (a) que en (b) y por tanto el error acumulado de la segmentación será mayor.

4.4.2. Resultados obtenidos para las secuencias del dataset PETS2009

En esta sección se analizan los resultados obtenidos para las secuencias **PETS2009-S1-L1-View001 13-57** y **PETS2009-S1-L1-View001 13-59**. Ambas secuencias se han tomado de una misma escena (mismo punto de vista) y con la misma cámara pero el *timestamp* de cada una es distinto.

En la tabla 4.5 se presentan los resultados obtenidos con la secuencia **PETS2009-S1-L1-View001** 13-57 para los sets de entrenamiento: 50:59, 131:140, 212:221 y 50:19:221, definidos como se explica en la sección anterior. Lo que más llama la atención en comparación con los resultados obtenidos para la secuencia UCSD, es que con el segmentador

	50:59		**131:140**		**212:221**		**50:19:221**		**valor medio**	
segmentador	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
p-f[a] **0.80 post**	**1.40**	3.43	**1.82**	**5.47**	**4.57**	**25.77**	1.84	5.30	**2.41**	**9.99**
p-f 0.85 post	1.60	4.26	2.45	8.60	5.06	31.11	1.55	3.64	2.66	11.90
p-f 0.80	1.54	3.66	2.57	9.44	5.30	34.10	1.58	3.72	2.75	12.73
p-f 0.75	1.38	**3.21**	2.99	12.11	5.21	33.30	1.51	3.42	2.77	13.01
p-f 0.70 post	2.12	6.68	2.56	10.76	5.71	44.93	1.70	4.75	3.02	16.78
p-f 0.85	1.86	5.02	2.51	10.03	5.81	40.99	1.98	5.84	3.04	15.47
p-f 0.70	1.93	6.11	3.53	16.10	5.46	36.85	1.66	4.32	3.14	15.84
p-f 0.90 post	2.17	7.48	2.44	9.40	6.03	43.45	2.26	8.23	3.22	17.14
p-f 0.90	2.08	6.35	2.64	10.81	6.93	57.10	3.01	13.93	3.67	22.05
p-f 0.75 post	1.44	3.29	1.91	6.85	17.04	342.11	**1.46**	**3.27**	5.46	88.88
f-f[b]	10.47	137.58	3.15	14.94	8.10	88.06	2.32	10.01	6.01	62.65

[a] **p-f: segmentador persona-fondo**

[b] **f-f: segmentador frente-fondo**

Cuadro 4.5: Errores de estimación del algoritmo implementado sobre la secuencia **PETS2009-S1-L1-View001** 13-57 con sets de entrenamiento 50:59, 131:140, 212:221 y 50:19:221.

persona-fondo las estimaciones son mucho más precisas (errores medios incluso menores que 2) mientas que con el segmentador frente-fondo el error ha aumentado considerablemente (errores medios de hasta 10). A continuación, se analizan las circunstancias que han dado lugar a este cambio en los resultados para uno y otro segmentador.

Atendiendo en primer lugar al segmentador persona-fondo, hay varias razones a las que se puede atribuir este cambio de tendencia. Por un lado, el tamaño de las personas es mayor que en los *frames* de la secuencia **UCSD**, por lo que se ajustan mucho mejor a la las escalas en las que se realiza la detección en el algoritmo de segmentación y por otro lado, la calidad de las imágenes es mejor, lo que hace que la detección de las partes del cuerpo sea más sencilla. En la figura 4.7 se presentan dos ejemplos de las máscaras de segmentación obtenidas para la versión que proporciona los mejores resultados, 0.80 post-procesada y una de las versiones que ofrece peores resultados, 0.90 sin post-procesado y un ejemplo de las máscaras del segmentador frente-fondo. Se puede ver claramente que el mejor resultado se obtiene con el segmentador persona-fondo con umbral 0.80 post-procesado ya que los segmentos incluyen a todas las personas de la escena y hay sólo 2 falsos positivos, ambos de un tamaño más pequeño que las personas de la escena. Una

desventaja del segmentador persona-fondo es que no es fácil elegir el umbral y la versión más adecuada (con o sin post-procesado), ya que depende, no solo de la secuencia, sino también del set de entrenamiento, como se puede ver en la tabla 4.5. Si nos fijamos en los errores obtenidos para 0.75 post-procesado, los errores son muy parecidos a los mejores obtenidos para todos los sets de entrenamiento excepto para el set 212:221, donde se obtienen los peores resultados.

Respecto al segmentador frente-fondo, a la luz de los resultados se puede afirmar que para los 3 primeros sets de entrenamiento, no se ha podido aprender la relación entre las características extraídas y el número de personas de la imagen, por lo que los errores de estimación son enormes. Esto se debe a que, como se puede ver en el ejemplo presentado en la figura 4.7, en la segmentación obtenida hay un alto porcentaje de los píxeles correspondientes a personas que no se detectan.

A continuación, en la tabla 4.6, se presentan los resultados de las estimaciones para la secuencia **PETS2009-S1-L1-View001 13-59** que, como se ha comentado antes, se ha capturado en el mismo escenario que la secuencia anterior y desde el mismo punto de vista, pero con un *timestamp* diferente.

Viendo los resultados se puede concluir que son bastante buenos con ambos segmentadores, segmentador persona-fondo y segmentador frente-fondo. Esta vez, salvo para el primer set de entrenamiento, el segmentador frente-fondo consigue las mejores puntuaciones, situándose el error por debajo de 1. En la figura 4.8 se presenta un ejemplo de la segmentación obtenida por el segmentador frente-fondo y el segmentador persona-fondo con umbral 0.80 sin post-procesado. Como se puede ver en el ejemplo, la máscara obtenida por el segmentador frente-fondo es mucho mejor que la del ejemplo de la secuencia **PETS2009-S1-L1-View001** 13-57 4.7. Esto se puede deber a que con la secuencia **PETS2009-S1-L1-View001** 13-57 el modelo de fondo no se ha podido inicializar correctamente.

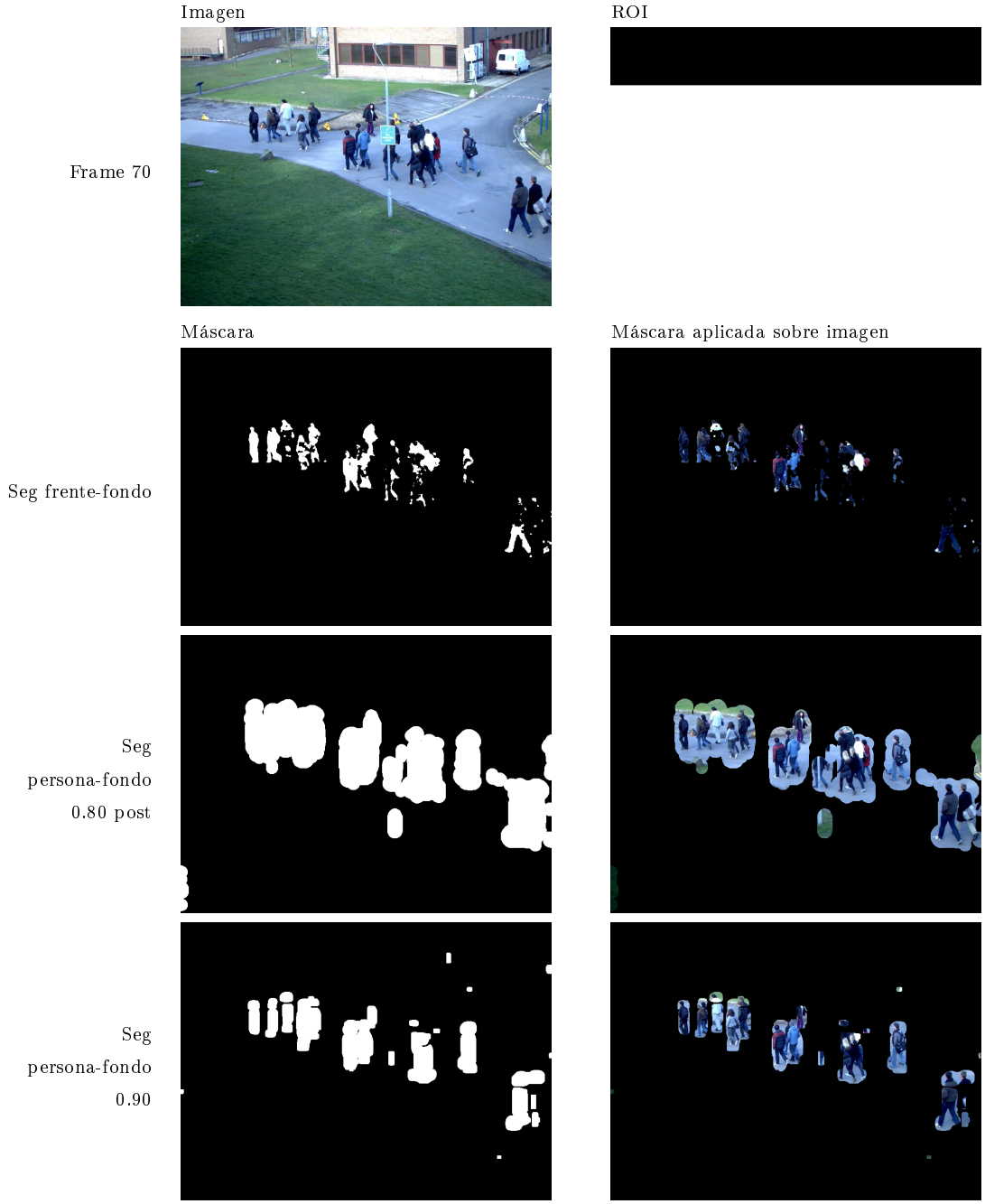

Figura 4.7: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **PETS2009-S1-L1-View001** 13-57.

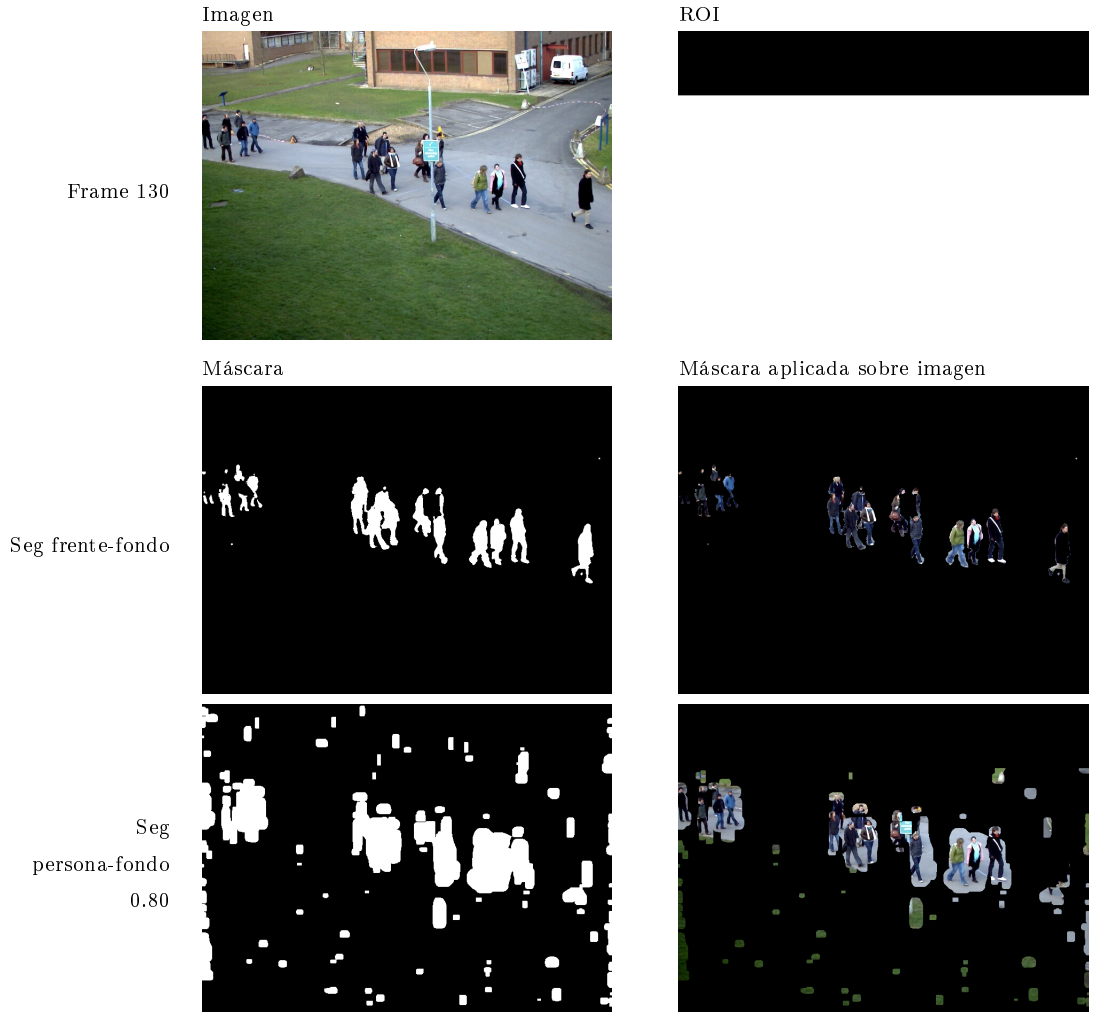

Figura 4.8: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **PETS2009-S1-L1-View001** 13-59.

	50:59		141:150		232:241		50:20:241		valor medio	
Seg	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
p-f[a] **0.80**	1.27	2.76	1.02	1.79	1.27	2.45	1.21	2.30	**1.19**	**2.33**
p-f 0.80 post	**1.21**	2.67	1.25	2.45	1.37	2.75	1.22	2.54	1.26	2.60
p-f 0.75 post	1.44	3.23	1.35	2.90	1.25	2.41	1.06	1.86	1.28	2.60
p-f 0.75	1.85	5.78	1.17	2.14	1.18	2.04	1.32	2.61	1.38	3.14
p-f 0.85 post	1.26	2.58	1.38	3.07	1.61	4.11	1.35	3.07	1.40	3.21
p-f 0.70	1.29	**2.47**	1.57	3.90	1.77	4.56	1.52	3.48	1.54	3.60
p-f 0.85	2.37	7.75	1.16	2.24	1.18	2.31	1.69	4.36	1.60	4.17
f-f[b]	5.09	33.21	**0.85**	**1.07**	**0.93**	**1.47**	**0.81**	**1.16**	1.92	9.23
p-f 0.70 post	1.83	5.40	1.68	4.50	3.11	16.86	1.53	3.88	2.04	7.66
p-f 0.90	3.83	18.12	1.69	4.93	1.59	4.18	1.68	4.53	2.20	7.94
p-f 0.90 post	3.77	18.92	2.99	13.74	2.12	7.55	2.69	11.52	2.89	12.93

[a] **p-f: segmentador persona-fondo**

[b] **f-f: segmentador frente-fondo**

Cuadro 4.6: Errores de estimación del algoritmo implementado sobre la secuencia **PETS2009-S1-L1-View001** 13-59 con sets de entrenamiento 50:59, 141:150, 232:241 y 50:20:241.

4.4.3. Resultados obtenidos para la secuencia PETS2006-S1-T1-View003

En esta sección se analizan los resultados obtenidos para la secuencia **PETS2006-S1-T1-View003**. Se trata de una secuencia muy simple, ya que la densidad máxima es de 5 personas. En las tablas 4.7 se muestran los resultados obtenidos.

Como se puede ver en la tabla, en el caso de la secuencia **PETS2006-S1-T1-View003** los errores obtenidos tanto con el segmentador frente-fondo como con casi todas las versiones y umbrales del segmentador persona-fondo son muy pequeños, siendo la mejor para todos los sets de entrenamiento la segmentación persona-fondo con umbral 0.85. En una secuencia como esta, donde ambos segmentadores funcionan perfectamente, el hecho de que las estimaciones de densidad obtenidas con el segmentador persona-fondo sean tan o más precisas que las del segmentador frente-fondo, constituye una evidencia de que partir de una máscara con contornos de personas más precisos, como la que se obtiene con el segmentador frente-fondo, no es determinante en el algoritmo de estimación de densidad. En la figura 4.9 se puede ver un ejemplo de la segmentación obtenida con el segmentador frente-fondo y con el segmentador persona fondo con umbral 0.85.

	5:14		58:67		112:121		5:11:121		valor medio	
segmentador	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
p-f[a] **0.85**	0.40	0.51	**0.39**	**0.51**	0.51	0.52	0.50	0.65	**0.45**	**0.55**
p-f 0.85 post	0.40	0.45	0.70	1.71	**0.41**	0.40	**0.42**	**0.53**	0.48	0.77
p-f 0.90 post	0.46	0.56	0.42	0.57	0.67	0.89	0.49	0.77	0.51	0.70
f-f[b]	**0.27**	**0.19**	0.49	0.61	0.82	2.41	0.50	0.81	0.52	1.00
p-f 0.90	0.50	0.65	0.50	0.79	0.61	0.61	0.52	0.75	0.53	0.70
p-f 0.75 post	0.64	0.78	0.71	0.93	**0.41**	**0.32**	0.49	0.65	0.56	0.67
p-f 0.80	0.53	0.65	0.58	1.15	0.70	0.89	0.56	1.17	0.59	0.97
p-f 0.80 post	0.46	0.53	0.61	1.08	0.60	0.66	0.75	1.73	0.61	1.00
p-f 0.70	0.76	1.13	0.67	1.09	0.49	0.38	0.51	0.63	0.61	0.81
p-f 0.70 post	0.59	0.57	0.64	0.89	0.58	0.87	0.65	1.11	0.62	0.86
p-f 0.75	0.56	0.59	0.78	1.52	0.47	0.43	0.67	1.05	0.62	0.90

[a] **p-f: segmentador persona-fondo**

[b] **f-f: segmentador frente-fondo**

Cuadro 4.7: Errores de estimación del algoritmo implementado sobre la secuencia **PETS2006-S1-T1-View003** con sets de entrenamiento 5:14, 58:67, 112:121 y 5:11:121.

4.4.4. Resultados obtenidos para las secuencias del dataset QUT

En esta sección se analizan los resultados obtenidos para las secuencias, **QUT-CameraA, QUT-CameraB** y **QUT-CameraC**.

En primer lugar, se presentan los errores de estimación obtenidos para la secuencia **QUT-CameraA** en la tabla 4.8. En este caso se trata de una secuencia con poca densidad de personas donde la que la principal dificultad radica en el ángulo de observación de la cámara con respecto al plano de tierra, que es muy elevado (ver *frame* de ejemplo en la figura 4.10). Los resultados demuestran que, como era de esperar el segmentador frente-fondo es capaz de lidiar mejor con esta situación, ya que para el segmentador persona-fondo es mucho más complicado detectar las partes del cuerpo bajo estas circunstancias. A pesar de ello, dada la poca densidad de los *frames*, no hay grandes diferencias en los errores de estimación con uno y otro tipo de segmentación. En la figura 4.10 se muestran ejemplos de las máscaras obtenidas al utilizar el segmentador frente-fondo y

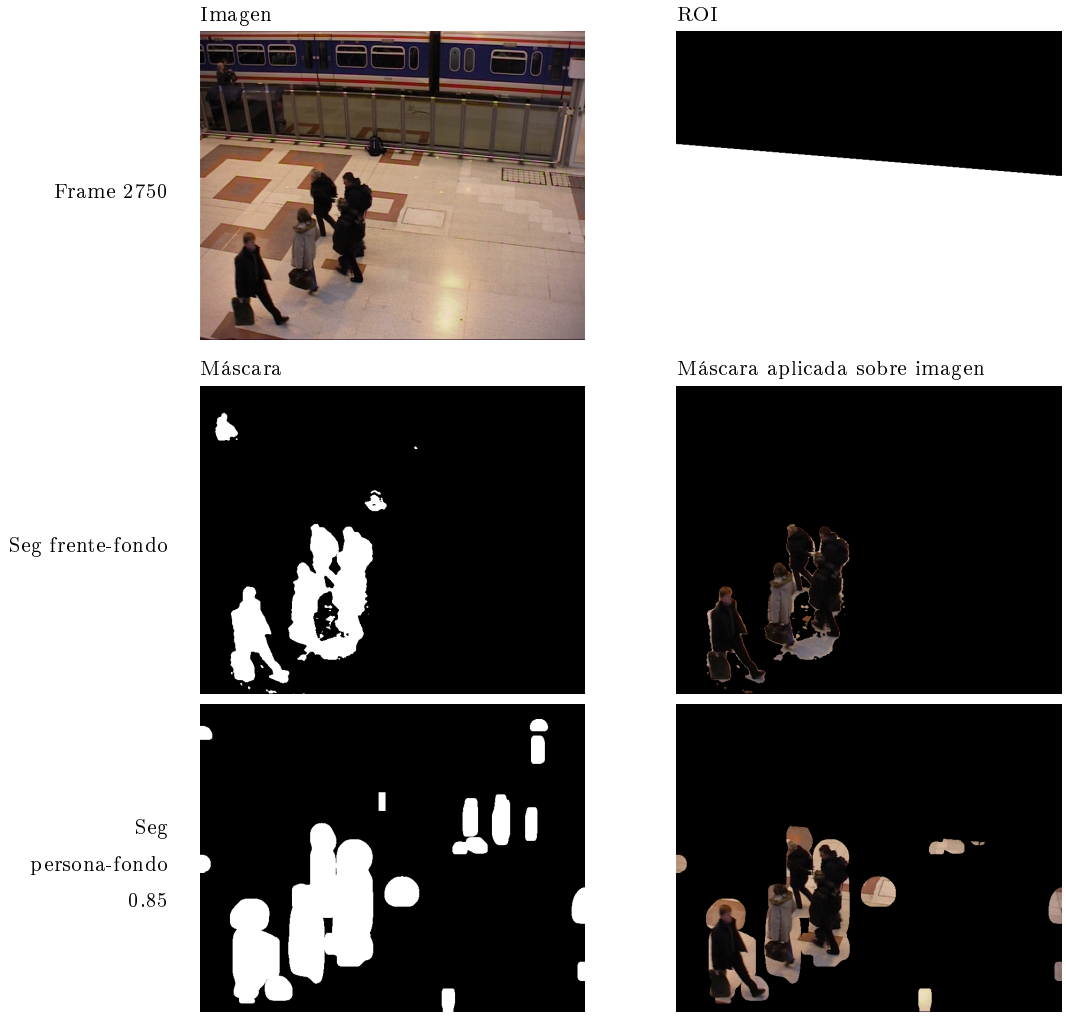

Figura 4.9: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **PETS2006-S1-T1-View003**.

	2:11		23:32		44:53		2:5:50		valor medio	
segmentador	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
f-f[a]	**0.50**	**0.59**	**0.48**	**0.46**	**0.70**	**1.10**	**0.42**	**0.42**	**0.53**	**0.64**
p-f[b] **0.85**	0.89	1.44	0.83	1.24	0.95	1.70	0.86	1.25	0.88	1.41
p-f 0.90	1.02	1.82	0.75	1.02	0.82	1.24	0.98	1.64	0.89	1.43
p-f 0.90 post	1.17	2.65	0.86	1.32	0.84	1.37	1.00	1.92	0.97	1.81
p-f 0.85 post	1.12	2.20	0.96	1.55	1.33	3.23	0.96	1.71	1.09	2.17
p-f 0.80	1.19	2.34	1.01	1.53	1.15	2.14	1.34	2.91	1.17	2.23
p-f 0.75	1.17	2.35	1.19	2.09	1.56	3.98	1.28	2.43	1.30	2.71
p-f 0.80 post	1.40	3.24	1.12	1.96	1.52	3.83	1.50	3.45	1.39	3.12
p-f 0.70	1.46	3.66	1.55	3.26	1.46	3.37	1.49	3.50	1.49	3.45
p-f 0.75 post	1.43	3.29	1.70	4.07	1.43	3.73	1.60	4.08	1.54	3.79
p-f 0.70 post	1.63	3.93	1.39	3.09	1.54	3.35	1.78	4.73	1.58	3.78

[a] **f-f: segmentador frente-fondo**

[b] **p-f: segmentador persona-fondo**

Cuadro 4.8: Errores de estimación del algoritmo implementado sobre la secuencia **QUT-CameraA** con sets de entrenamiento 2:11, 23:32, 44:53 y 2:5:50.

las versiones y umbrales del segmentador persona-fondo que mejores resultados ofrecen en la estimación de densidad: umbral 0.85 sin post-procesado y 0.70 post-procesado. Se puede ver que en la máscara del segmentador frente-fondo se incluyen las sombras de las personas, lo cual no ha afectado apenas a los resultados ya que el tamaño de los segmentos aumenta de manera más o menos proporcional debido a que todas las personas en la escena dejan una sombra similar.

En la tabla 4.9 se presentan los resultados para la secuencia **QUT-CameraB**. Esta secuencia es muy similar a la anterior, solo que en este caso la densidad de las imágenes es muy elevada, llegando incluso a 23 personas. Por este motivo, ambos tipos de segmentación ofrecen peores resultados que para la secuencia anterior, pero comparativamente el segmentador frente-fondo sigue siendo superior al segmentador persona-fondo. En la figura 4.11 se muestran ejemplos de la segmentación con el segmentador frente-fondo y el segmentador persona-fondo con umbral 0.85 y 0.90 con post-procesado.

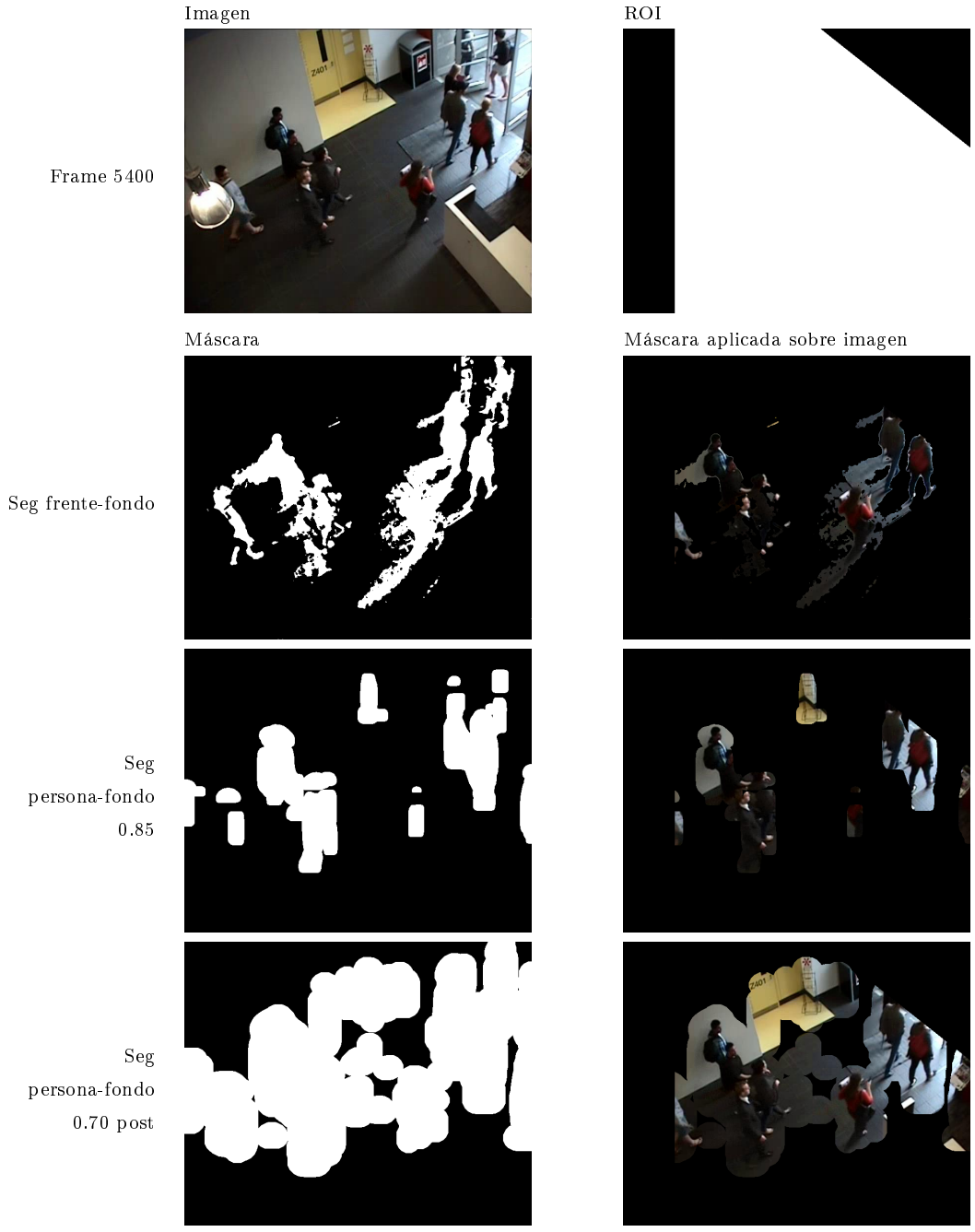

Figura 4.10: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **QUT-CameraA**.

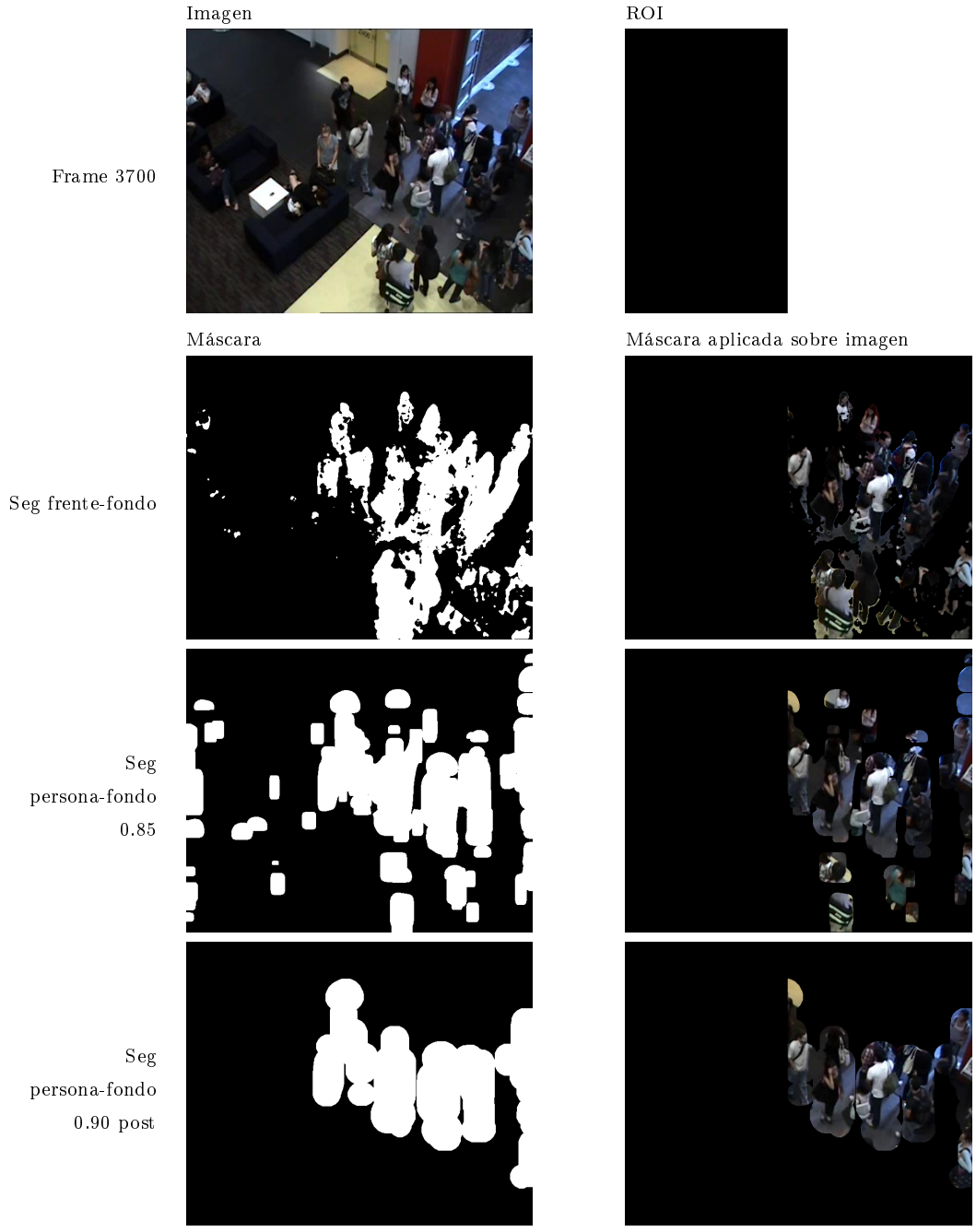

Figura 4.11: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **QUT-CameraB**.

	2:11		21:30		41:50		2:5:50		valor medio	
segmentador	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
f-f[a]	**2.80**	**15.20**	**2.29**	**9.04**	2.50	9.11	**1.78**	**4.86**	**2.34**	**9.55**
p-f[b] **0.85**	4.51	39.08	3.01	18.65	2.70	11.52	2.29	8.49	3.13	19.43
p-f 0.85 post	5.62	57.30	2.88	17.20	2.81	14.18	2.17	7.09	3.37	23.94
p-f 0.75	5.18	55.20	2.96	15.16	3.23	15.40	2.63	9.10	3.50	23.72
p-f 0.80	5.07	51.78	2.73	12.87	3.55	17.79	2.67	9.97	3.51	23.10
p-f 0.80 post	5.90	63.74	3.32	23.97	2.58	9.86	2.40	8.47	3.55	26.51
p-f 0.75 post	5.63	62.32	4.02	36.27	2.93	11.39	1.87	4.94	3.61	28.73
p-f 0.90	5.26	50.12	3.42	23.93	3.00	16.35	2.96	16.10	3.66	26.62
p-f 0.70 post	5.80	68.88	5.18	55.60	**2.42**	**8.75**	2.27	7.99	3.92	35.31
p-f 0.70	5.58	63.51	4.25	41.20	3.80	20.30	2.77	11.98	4.10	34.25
p-f 0.90 post	5.95	63.32	4.00	36.43	3.46	24.10	3.30	25.22	4.18	37.27

[a] **f-f: segmentador frente-fondo**

[b] **p-f: segmentador persona-fondo**

Cuadro 4.9: Errores de estimación del algoritmo implementado sobre la secuencia **QUT-CameraB** con sets de entrenamiento 2:11, 23:32, 44:53 y 2:5:50.

A continuación, en la tabla 4.10 se muestran los resultados obtenidos para la secuencia **QUT-CameraC**. Se trata de una secuencia parecida a las dos anteriores, por lo que los resultados una vez más son mejores utilizando el segmentador frente-fondo que el segmentador fondo-persona. En la figura 4.12 se pueden ver ejemplos de ambos tipos de segmentación.

4.4.5. Resultados obtenidos para las secuencias del dataset TUD

En esta sección se analizan los resultados obtenidos para las secuencias **TUD-campus** y **TUD-crossing**. Ambas secuencias tienen densidades máximas muy bajas (7 personas en el caso de **TUD-campus** y 11 personas en el caso de **TUD-crossing**) y el ángulo de orientación de la cámara con respecto al plano de tierra es muy pequeño, lo que se espera, favorezca al segmentador persona-fondo. Además, en la escena de la secuencia **TUD-crossing** hay coches en movimiento, lo que deberá afectar principalmente al segmentador frente-fondo que los detectará como parte del frente de la imagen.

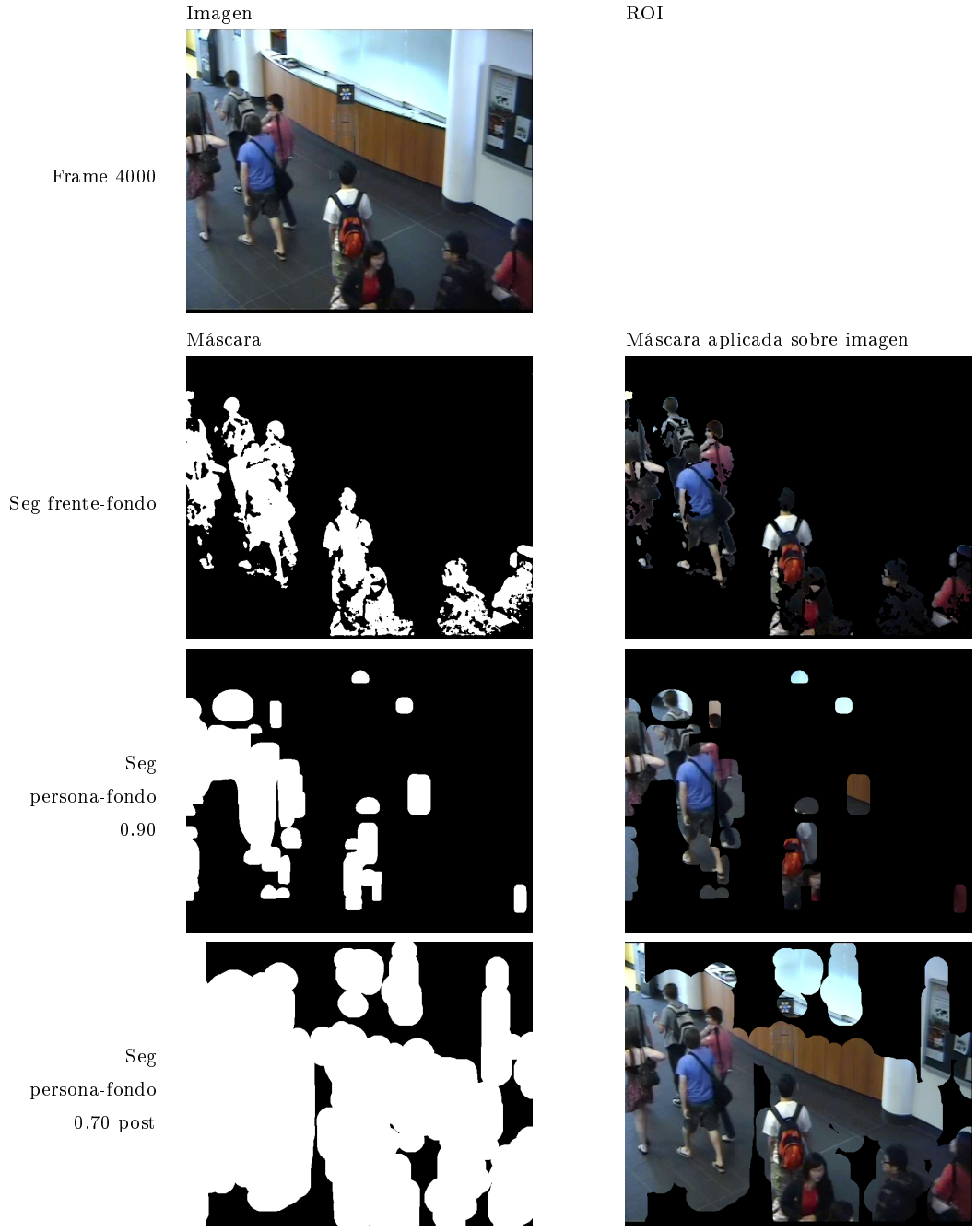

Figura 4.12: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **QUT-CameraC**.

	2:11		21:30		41:50		2:5:50		valor medio	
segmentador	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
f-f[a]	**0.68**	**0.81**	**0.76**	**0.90**	**0.43**	**0.39**	**0.47**	**0.47**	**0.58**	**0.64**
p-f[b] **0.90**	1.60	3.94	1.64	4.13	1.30	2.46	1.12	2.01	1.42	3.13
p-f 0.90 post	1.93	5.70	1.44	3.27	1.36	2.96	1.21	2.39	1.49	3.58
p-f 0.85	1.47	3.30	2.26	8.06	1.07	1.91	1.22	2.33	1.50	3.90
p-f 0.80	1.49	3.26	2.00	6.56	1.37	2.64	1.37	2.71	1.56	3.79
p-f 0.75	1.51	3.58	1.88	6.07	1.61	3.98	1.48	3.53	1.62	4.29
p-f 0.85 post	1.98	6.27	1.56	4.25	1.65	3.79	1.41	3.17	1.65	4.37
p-f 0.80 post	2.12	6.98	1.88	6.41	1.56	3.60	1.36	2.75	1.73	4.94
p-f 0.70	1.95	5.42	2.61	9.75	1.84	5.16	1.67	4.35	2.02	6.17
p-f 0.75 post	1.92	5.84	2.57	10.89	2.07	5.90	1.54	3.76	2.03	6.60
p-f 0.70 post	2.31	8.62	2.52	9.32	1.41	3.19	2.25	7.18	2.12	7.08

[a] **f-f: segmentador frente-fondo**

[b] **p-f: segmentador persona-fondo**

Cuadro 4.10: Errores de estimación del algoritmo implementado sobre la secuencia **QUT-CameraC** con sets de entrenamiento 2:11, 23:32, 44:53 y 2:5:50.

En la tabla 4.11 se muestran los errores de estimación para la secuencia **TUD-campus**. Tal y como se esperaba, los resultados con el segmentador persona-fondo son muy buenos, obteniendo errores muy pequeños. Los errores en la estimación con el segmentador frente-fondo son ligeramente superiores que los mejores resultados del segmentador persona-fondo, aunque también bajos. En la figura 4.13 se muestran ejemplos de las segmentaciones realizadas.

Los resultados de las evaluaciones sobre la secuencia **TUD-crossing** se muestran en la tabla 4.12. De acuerdo a lo esperado, el segmentador persona-fondo es capaz de lidiar mejor con una escena como esta, en la que aparecen coches en movimiento, que el segmentador frente-fondo, ya que este detecta los píxeles de los coches como parte del frente. En las figura 4.14 y 4.15 se muestran dos ejemplos de las máscaras de segmentación de ambos tipos de segmentadores donde se puede apreciar claramente esta situación.

	2:11		32:41		62:71		2:7:70		valor medio	
segmentador	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
p-f[a] **0.70 post**	**0.50**	**0.39**	**0.50**	**0.50**	0.62	0.57	**0.49**	**0.40**	**0.53**	**0.46**
p-f 0.70	0.71	0.71	0.70	0.80	**0.54**	**0.42**	0.65	0.62	0.65	0.64
p-f 0.80 post	0.55	0.47	0.67	0.67	0.87	1.25	0.62	0.58	0.67	0.74
p-f 0.75 post	0.75	0.87	0.77	0.96	0.56	0.50	0.64	0.61	0.68	0.74
p-f 0.85 post	0.86	1.05	0.74	0.78	0.67	0.71	0.60	0.59	0.72	0.78
p-f 0.75	1.06	1.51	0.82	1.04	0.61	0.54	0.54	0.46	0.76	0.89
f-f[b]	0.79	0.84	1.04	1.58	0.77	0.87	0.50	0.37	0.78	0.91
p-f 0.80	0.95	1.27	0.84	1.08	0.75	0.91	0.60	0.61	0.78	0.97
p-f 0.85	0.99	1.37	0.80	0.96	0.79	0.91	0.64	0.66	0.81	0.98
p-f 0.90	0.99	1.49	0.94	1.35	0.63	0.61	0.70	0.71	0.82	1.04
p-f 0.90 post	0.97	1.43	0.97	1.38	0.87	1.21	0.77	0.94	0.90	1.24

[a] **p-f: segmentador persona-fondo**

[b] **f-f: segmentador frente-fondo**

Cuadro 4.11: Errores de estimación del algoritmo implementado sobre la secuencia **TUD-Campus** con sets de entrenamiento 2:11, 32:41, 62:71 y 2:7:70.

	2:11		97:106		192:201		2:20:201		valor medio	
segmentador	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
p-f[a] **0.85 post**	0.64	0.67	**0.70**	0.83	**0.75**	**0.95**	**0.72**	**0.86**	**0.70**	**0.83**
p-f 0.80 post	1.04	1.60	1.12	2.09	0.95	1.46	0.76	0.90	0.97	1.51
p-f 0.90 post	0.63	0.60	**0.70**	**0.72**	1.75	4.44	0.80	1.08	0.97	1.71
p-f 0.85	**0.55**	**0.44**	**0.70**	0.88	1.97	5.78	0.85	1.23	1.02	2.08
p-f 0.90	0.85	1.16	0.83	1.14	2.14	6.45	0.74	0.93	1.14	2.42
p-f 0.80	0.79	1.03	0.91	1.28	2.12	7.13	0.83	1.14	1.16	2.65
p-f 0.75 post	1.10	1.85	1.29	2.29	1.32	2.55	1.13	1.90	1.21	2.15
f-f[b]	0.91	1.29	1.87	6.15	1.77	4.67	0.95	1.35	1.37	3.36
p-f 0.70 post	1.52	3.91	2.33	6.79	0.79	1.05	0.89	1.21	1.38	3.24
p-f 0.70	1.07	1.59	1.71	3.60	2.46	9.33	0.99	1.44	1.55	3.99
p-f 0.75	0.92	1.26	1.55	3.08	3.06	15.17	0.87	1.18	1.60	5.17

[a] **p-f: segmentador persona-fondo**

[b] **f-f: segmentador frente-fondo**

Cuadro 4.12: Errores de estimación del algoritmo implementado sobre la secuencia **TUD-Crossing** con sets de entrenamiento 2:11, 97:106, 192:201 y 2:20:201.

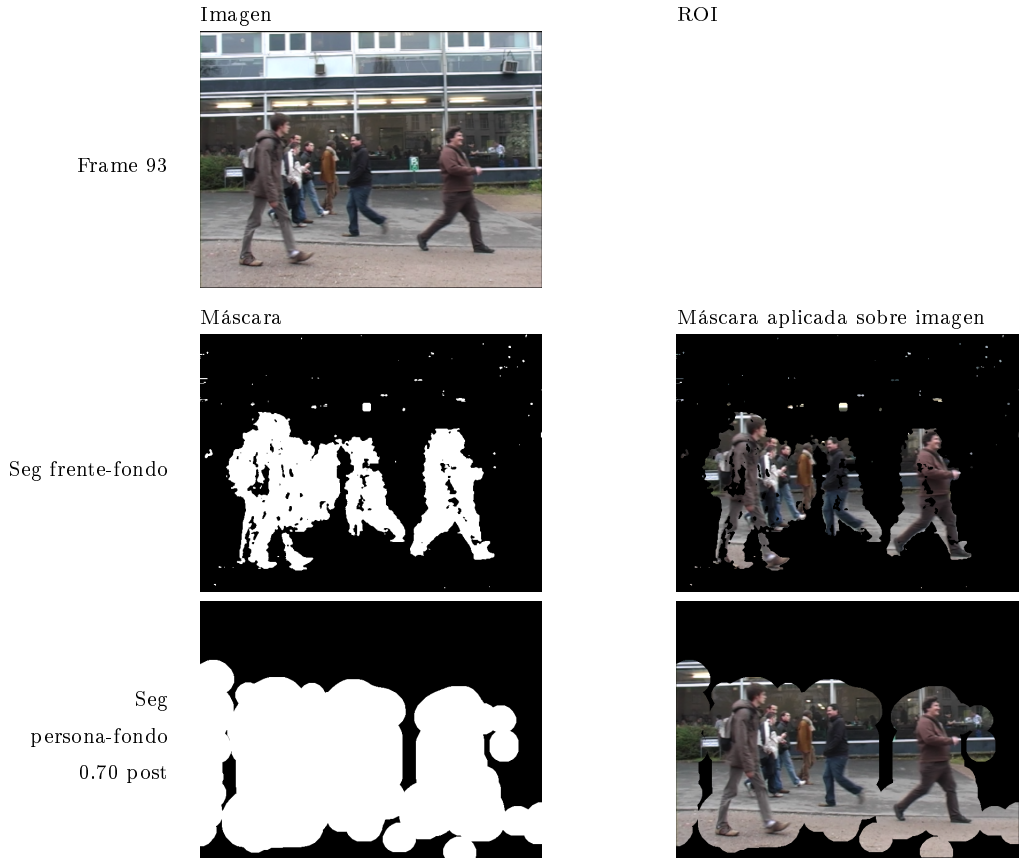

Figura 4.13: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **TUD-campus**.

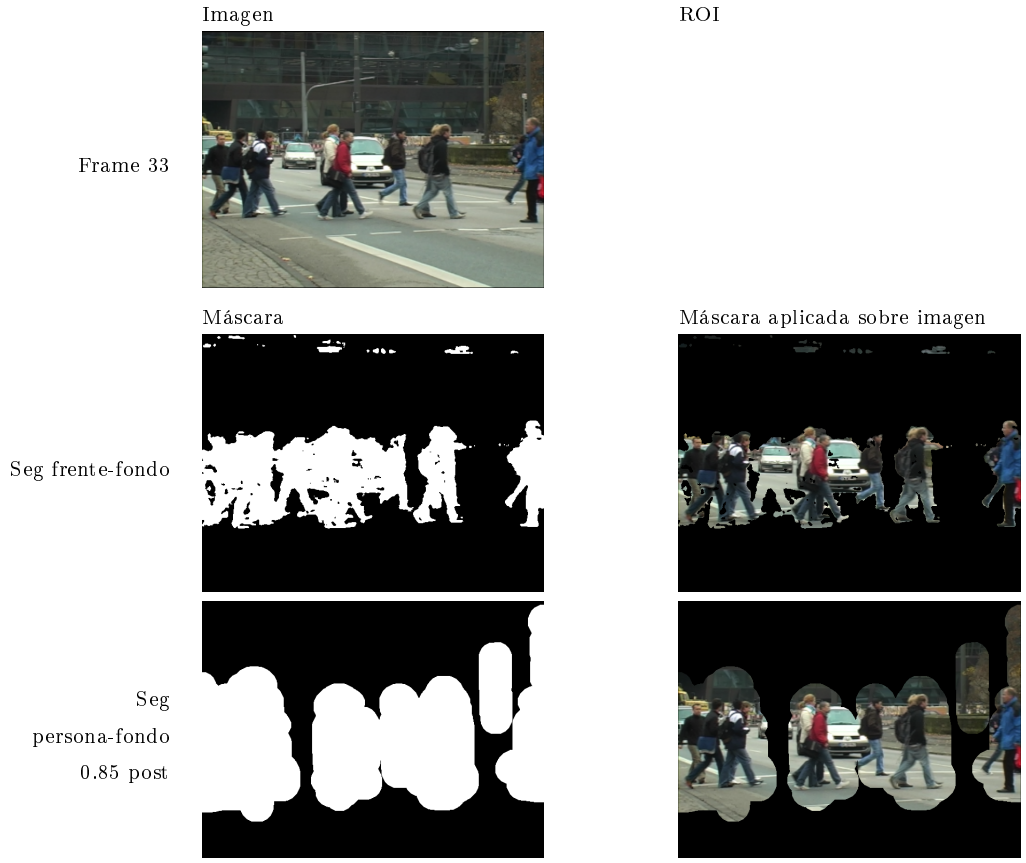

Figura 4.14: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **TUD-crossing**, ejemplo1.

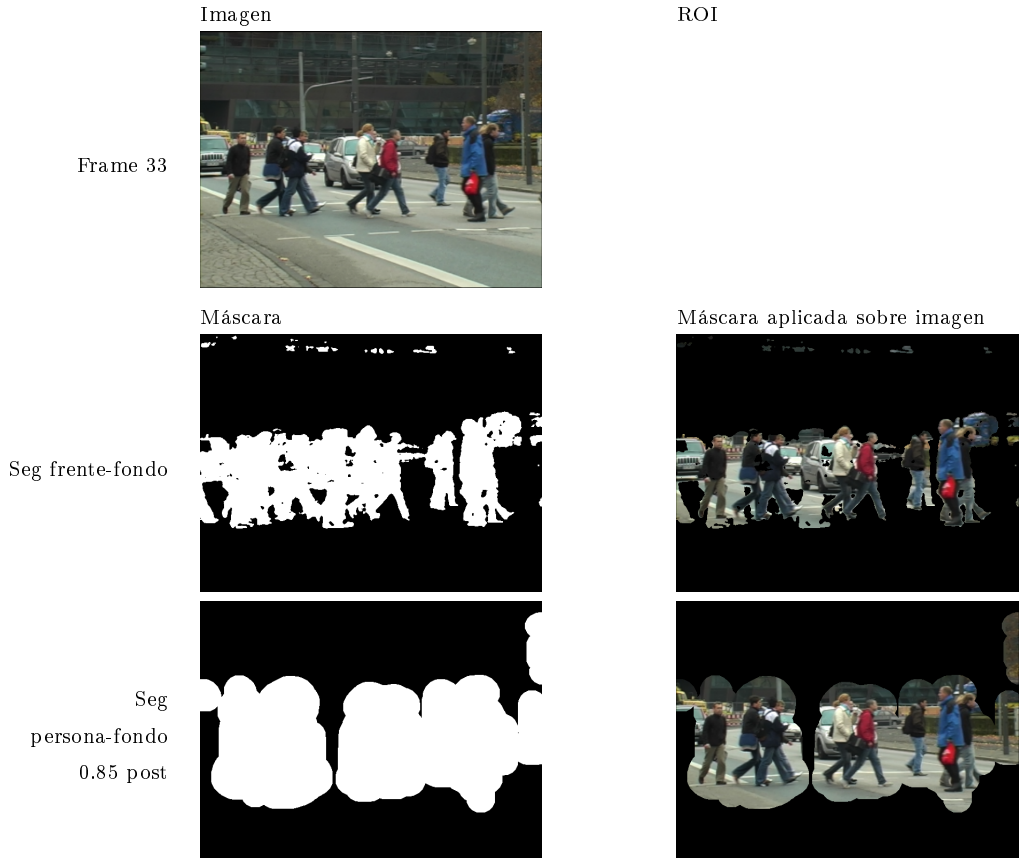

Figura 4.15: Diferencias segmentador frente-fondo y segmentador persona-fondo en secuencia **TUD-crossing**, ejemplo2.

4.5. Conclusiones

Como se ha podido ver a lo largo de toda la sección gracias a las evaluaciones realizadas sobre distintas secuencias, tanto el segmentador frente-fondo, como el segmentador persona-fondo presentan ventajas y desventajas que les hacen más apropiados para unas secuencias que para otras. Por tanto, no se puede afirmar de manera absoluta que uno de los dos es mejor para realizar la segmentación dentro del algoritmo de estimación de densidad de persona, sino que se debe elegir el que mejor se adapte a las características concretas de la escena o combinarlos para aprovechar las ventajas de cada uno. A continuación, se resumen sus características más relevantes de cara a su utilización como parte del algoritmo de estimación de densidad de personas.

Comenzando por el segmentador frente-fondo, se ha comprobado que es la opción más acertada cuando se trata de escenas con mucha densidad de personas donde puede haber grandes oclusiones o alguna otra circunstancia que dificulte la detección de las personas, como por ejemplo, que el ángulo de observación de la cámara con respecto al plano de tierra sea muy grande, etc. También ofrece mejores resultados cuando las personas se ven muy pequeñas o cuando la imagen no tiene mucha calidad como consecuencia de la baja resolución de la cámara. Sin embargo, hay otras situaciones en las que este algoritmo supondrá una fuente de error considerable para la estimación de densidad. La primera de ellas, es cuando los *frames* de la frecuencia no tienen suficiente contraste o nitidez, lo que ocasionará que las intensidades de los píxeles del frente y del fondo sean similares y que no se detecte bien el movimiento. Otra circunstancia que afecta enormemente a los resultados de este algoritmo, es el hecho de que en la escena haya otros objetos en movimiento que se puedan detectar como parte del frente, como por ejemplo coches o sombras de las propias personas. Además, en algunas secuencias, el modelo de fondo no se inicializa correctamente debido a diversos factores, como por ejemplo que haya muchas personas en la escena desde el comienzo de la secuencia.

En el caso del segmentador persona-fondo, hay varias razones por las que en ocasiones su utilización resulta menos conveniente. Además de las ya comentadas como parte de las ventajas del segmentador frente-fondo, se tiene, por un lado, que el tiempo de ejecución necesario para realizar la segmentación es mucho mayor que con segmentador frente-fondo y, por otro lado, que se debe determinar el umbral y la versión óptima (post-procesada

o no) para la escena en cuestión, ya que las pruebas realizadas demuestran que es difícil establecer alguna regla para conocer de antemano cuáles serán los mejores. No obstante, al estar basado en la detección de personas, en cualquier situación como la que se menciona arriba, donde el algoritmo de segmentación frente-fondo no pueda detectar el movimiento, o la detección de todos los objetos en movimiento implique añadir al frente de la imagen objetos que no deban ser contados como personas, el algoritmo de segmentación persona-fondo es una alternativa mejor para la estimación de densidad de personas. Otra de las ventajas de este algoritmo, es que podría utilizarse con una cámara móvil, mientras que con el segmentador frente-fondo sería necesario primero corregir el movimiento de la cámara.

A todo lo dicho hasta el momento se puede añadir que aunque en líneas generales el algoritmo de segmentación frente-fondo es preferible para realizar la estimación de densidad de personas, salvo en circunstancias muy concretas, la segmentación persona-fondo puede ser muy útil como preprocesado o en combinación con un algoritmo de segmentación frente-fondo para aquellos casos en los que ésta presente dificultades. Por último, resulta importante destacar que la segmentación frente-fondo y concretamente los algoritmos **BGS,** se han estado estudiando durante muchos años, por lo que los algoritmos utilizados son cada vez más complejos y obtienen segmentaciones más precisas. En cambio, la segmentación persona-fondo ha aparecido recientemente y no se ha estudiado apenas, lo que sugiere que con el esfuerzo de la comunidad científica en este tema, se podrían corregir muchas de sus deficiencias y aumentar su precisión, lo que sin duda la haría más atractiva para la estimación de densidad de personas.

5

Conclusiones y trabajo futuro

5.1. Conclusiones

Como bien se ha expuesto en la sección 1.1 del capítulo 1, la estimación de densidad de personas se ha convertido en una de las tareas primordiales en la monitorización de personas en lugares públicos, lo que ha motivado la aparición de un gran número de algoritmos de estimación de densidad. Dado que una parte importante de estos algoritmos, incluyen una etapa previa de segmentación de los *frames* de la secuencia, para posteriormente extraer ciertas características del frente de la imagen, se ha fijado como objetivo principal del proyecto la implementación de un algoritmo de densidad de personas y la comparación de los resultados obtenidos utilizando dos tipos de segmentadores diferentes: un segmentador frente-fondo y un segmentador persona-fondo. Para ello, en la sección 1.2, se han marcado previamente una serie de hitos que se han ido alcanzando a lo largo del proyecto. En la presente sección, se presenta un resumen de toda esta evolución y de las conclusiones finales alcanzadas.

En primer lugar, en la sección 2.2 del capítulo 2, se ha realizado un estudio del Estado del Arte de estimación de densidad de personas y se ha elegido un algoritmo basado en segmentación atendiendo principalmente a sus resultados. Además, en la sección 2.3.1

se ha analizado el Estado del Arte de la segmentación, específicamente los algoritmos de sustracción de fondo (**BGS**) y se ha elegido también un algoritmo de este tipo para segmentar las imágenes como parte del proceso de estimación de densidad. En la sección 2.3.2, se ha estudiado el único algoritmo de segmentación persona-fondo desarrollado hasta el momento.

En el capítulo 3, se han explicado detalladamente cada una de las etapas del algoritmo de estimación de densidad de personas y su implementación. Como parte de las estapas del algoritmo, en la secciones 3.2.1 y 3.2.2 del capítulo 3, se ha explicado el funcionamiento de los algoritmos de segmentación frente-fondo y persona-fondo respectivamente.

Por último, en el capítulo 4, se evalúa la precisión del algoritmo y se presentan los resultados obtenidos con ambos segmentadores junto a los resultados obtenidos por el algoritmo original y otros algoritmos similares del Estado del Arte. A continuación, se presentan todas las evaluaciones realizadas para comparar los resultados de la estimación utilizando ambos segmentadores. En el caso del algoritmo de segmentación persona-fondo, se comparan a la vez las estimaciones obtenidas con distintos umbrales y dos versiones distintas del algoritmo.

Atendiendo a los resultados, podemos concluir que se han alcanzado los objetivos del proyecto, ya que el algoritmo de estimación de densidad de personas implementado ha permitido comparar el uso de ambos tipos de segmentacción para la estimación de densidad y se ha demostrado que aunque todos los algoritmos de estimación de densidad basados en la extracción previa del fondo de la imagen emplean segmentadores frente-fondo, la segmentación persona-fondo, puede aportar ciertas ventajas en escenarios concretos, tal y como se ha explicado a lo largo de toda la sección 4.4 y se ha resumido en la sección 4.5.

5.2. Trabajo futuro

Existen numerosas líneas de investigación que a pesar de que no fomaban parte de los objetivos de este proyecto, podrían contribuir de manera notable a la mejora de los resultados de estimación de densidad de personas.

1. En primer lugar, sería interesante fusionar los dos segmentadores para obtener una

única máscara que integrase las ventajas de ambos.

2. Otra posible mejora se podría conseguir utilizando otros algoritmos de modelado de fondo y de segmentación persona-fondo.

3. Dado que el algoritmo de segmentación frente-fondo utilizado en este caso es un algoritmo de modelado de fondo (**BGS**), otra posible alternativa sería emplear otro tipo de algoritmo para comparar los resultados.

4. Además de las modificaciones orientadas a mejorar la etapa de segmentación, se podrían realizar variaciones sobre otras etapas de la estimación de densidad, como por ejemplo, en la extracción de características, emplear características distintas a las utilizadas en este proyecto o probar otros algoritmos de regresión.

Bibliografía

[1] D. Baltieri, R. Vezzani, and R. Cucchiara. Fast background initialization with recursive hadamard transform. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 165–171, Aug 2010.

[2] Hamidreza Baradaran Kashani, Seyed Alireza Seyedin, and Hadi Sadoghi Yazdi. A novel approach in video scene background estimation. *International Journal of Computer Theory and Engineering*, 2(2):274–282, April 2010.

[3] O. Barnich and M. Van Droogenbroeck. Vibe: A powerful random technique to estimate the background in video sequences. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 945–948, April 2009.

[4] T. Bouwmans. *Background subtraction for visual surveillance: A fuzzy approach.* 2012.

[5] Thierry Bouwmans. Subspace learning for background modeling: A survey. 2009.

[6] Thierry Bouwmans. Recent advanced statistical background modeling for foreground detection-a systematic survey. 2011.

[7] Thierry Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, (0):31 – 66, 2014.

[8] G.J. Brostow and R. Cipolla. Unsupervised Bayesian detection of independent motion in crowds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 594–601, 2006.

[9] Darren E. Butler, V. Michael Bove, Jr., and Sridha Sridharan. Real-time adaptive foreground/background segmentation. *EURASIP J. Appl. Signal Process.*, 2005:2292–2304, January 2005.

[10] Emmanuel J. Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

[11] A.B. Chan, Z.-S.J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7, 2008.

[12] Remy Chang, T. Gandhi, and M.M. Trivedi. Vision modules for a multi-sensory bridge monitoring approach. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 971–976, Oct 2004.

[13] P. Chiranjeevi and S. Sengupta. Detection of moving objects using fuzzy correlogram based background subtraction. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pages 255–259, Nov 2011.

[14] P. Chiranjeevi and S. Sengupta. New fuzzy texture features for robust detection of moving objects. *Signal Processing Letters, IEEE*, 19(10):603–606, Oct 2012.

[15] P. Chiranjeevi and S. Sengupta. Robust detection of moving objects in video sequences through rough set theory framework. *Image and Vision Computing*, 30(11):829 – 842, 2012.

[16] Siu-Yeung Cho, T.W.S. Chow, and Chi-Tat Leung. A neural-based crowd estimation by hybrid global learning algorithm. 29(4):535–541, 1999.

[17] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht. Neural network approach to background modeling for video object segmentation. *Neural Networks, IEEE Transactions on*, 18(6):1614–1627, Nov 2007.

[18] Ciprian David, Vasile Gui, and Florin Alexa. Foreground/background segmentation with learned dictionary. In *Proceedings of the 3rd International Conference on Applied Mathematics, Simulation, Modelling, Circuits, Systems and Signals*, ASMCSS'09, pages 197–201, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS).

[19] A.C. Davies, Jia Hong Yin, and S.A. Velastin. Crowd monitoring using image processing. *Electronics & Communication Engineering Journal*, 7(1):37–47, 1995.

[20] Jianwei Ding, Min Li, Kaiqi Huang, and Tieniu Tan. Modeling complex scenes for accurate moving objects segmentation. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision, ACCV 2010*, volume 6493 of *Lecture Notes in Computer Science*, pages 82–94. Springer Berlin Heidelberg, 2011.

[21] F. El Baf, T. Bouwmans, and B. Vachon. Foreground detection using the choquet integral. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS '08. Ninth International Workshop on*, pages 187–190, May 2008.

[22] F. El Baf, T. Bouwmans, and B. Vachon. A fuzzy approach for background subtraction. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2648–2651, 2008.

[23] F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 60–65, June 2009.

[24] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In David Vernon, editor, *Computer Vision, ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 751–767. Springer Berlin Heidelberg, 2000.

[25] R.H. Evangelio, M. Patzold, I. Keller, and T. Sikora. Adaptively splitted gmm with feedback improvement for the task of background subtraction. *Information Forensics and Security, IEEE Transactions on*, 9(5):863–874, May 2014.

[26] R.H. Evangelio and T. Sikora. Complementary background models for the detection of static and moving objects in crowded environments. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 71–76, Aug 2011.

[27] Diana Farcas, Cristina Marghes, and Thierry Bouwmans. Background subtraction via incremental maximum margin criterion: a discriminative subspace approach. *Machine Vision and Applications*, 23(6):1083–1101, 2012.

[28] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. 32(9):1627–1645, 2010.

[29] Dongfa Gao, Zhuolin Jiang, and Ming Ye. A new approach of dynamic background modeling for surveillance information. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 1, pages 850–855, Dec 2008.

[30] Tao Gao, Zheng-guang Liu, Wen-chun Gao, and Jun Zhang. A robust technique for background subtraction in traffic video. In Mario Koppen, Nikola Kasabov, and George Coghill, editors, *Advances in Neuro-Information Processing*, volume 5507 of *Lecture Notes in Computer Science*, pages 736–744. Springer Berlin Heidelberg, 2009.

[31] A. Garcia-Martin, A. Cavallaro, and J.M. Martinez. People-background segmentation with unequal error cost. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 157–160, 2012.

[32] Y. Guan. Wavelet multi-scale transform based foreground segmentation and shadow elimination. *Open Signal Processing Journal*, 1:1–6, 2008.

[33] Ling Guo and Ming hui Du. Student's t-distribution mixture background model for efficient object detection. In *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on*, pages 410–414, Aug 2012.

[34] TomS.F. Haines and Tao Xiang. Background subtraction with dirichlet processes. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision, ECCV 2012*, volume 7575 of *Lecture Notes in Computer Science*, pages 99–113. Springer Berlin Heidelberg, 2012.

[35] B. Han, D. Comaniciu, and Larry S. Davis. Sequential kernel density approximation through mode propagation: applications to background modeling. *Proc. ACCV*, 2004.

[36] Bohyung Han and Ramesh Jain. Real-time subspace-based background modeling using multi-channel data. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Nikos Paragios, Syeda-Mahmood Tanveer, Tao Ju, Zicheng Liu, Sabine Coquillart, Carolina Cruz-Neira, Torsten Muller, and Tom Malzbender, editors,

Advances in Visual Computing, volume 4842 of *Lecture Notes in Computer Science*, pages 162–172. Springer Berlin Heidelberg, 2007.

[37] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38–43, June 2012.

[38] Ya-Li Hou and G.K.H. Pang. People counting and human detection in a challenging situation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 41(1):24–33, 2011.

[39] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *International Conference on Machine Learning, ICML 2009*, 2009.

[40] J.C.S. Jacques Junior, S. Raupp Musse, and C.R. Jung. Crowd analysis using computer vision techniques. 27(5):66–77, 2010.

[41] AnandSingh Jalal and Vrijendra Singh. A robust background subtraction approach based on daubechies complex wavelet transform. In Ajith Abraham, Jaime Lloret Mauri, JohnF. Buford, Junichi Suzuki, and SabuM. Thampi, editors, *Advances in Computing and Communications*, volume 191 of *Communications in Computer and Information Science*, pages 516–524. Springer Berlin Heidelberg, 2011.

[42] M.J. Jones and D. Snow. Pedestrian detection using boosted features over many frames. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, 2008.

[43] K. P. Karmann and A. Brandt. Moving object recognition using an adaptive background memory. In *Time-Varying Image Processing and Moving Object Recognition, V. Cappellini, ed. 2*, pages 289–307. Elsevier Science Publishers B.V., 1990.

[44] Hansung Kim, Ryuuki Sakamoto, Itaru Kitahara, Tomoji Toriyama, and Kiyoshi Kogure. Robust foreground extraction technique using gaussian family model and multiple thresholds. In Yasushi Yagi, SingBing Kang, InSo Kweon, and Hongbin Zha, editors, *Computer Vision, ACCV 2007*, volume 4843 of *Lecture Notes in Computer Science*, pages 758–768. Springer Berlin Heidelberg, 2007.

[45] Kyungnam Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 5, pages 3061–3064 Vol. 5, Oct 2004.

[46] Wonjun Kim and Changick Kim. Background subtraction for dynamic texture scenes using fuzzy color histograms. *Signal Processing Letters, IEEE*, 19(3):127–130, March 2012.

[47] Dan Kong, D. Gray, and Hai Tao. A viewpoint invariant approach for crowd counting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1187–1190, 2006.

[48] M.T. Gopala Krishna, V.N. Manjunath Aradhya, M. Ravishankar, and D.R. Ramesh Babu. Lopp: Locality preserving projections for moving object detection. *Procedia Technology*, 4(0):624 – 628, 2012. 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) on February 25 - 26, 2012.

[49] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 878–885, 2005.

[50] Victor S. Lempitsky and Andrew Zisserman. Learning to count objects in images. In *NIPS'10*, pages 1324–1332, 2010.

[51] Jing Li, Junzheng Wang, and Wei Shen. Moving object detection in framework of compressive sampling. *Systems Engineering and Electronics, Journal of*, 21(5):740–745, Oct 2010.

[52] Liyuan Li, Weimin Huang, I.Y.-H. Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. 13(11):1459–1472, 2004.

[53] Horng-Horng Lin, Tyng-Luh Liu, and Jen-Hui Chuang. A probabilistic svm approach for background scene initialization. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages 893–896 vol.3, June 2002.

[54] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 31(6):645–654, 2001.

[55] Zhou Liu, Wei Chen, Kaiqi Huang, and Tieniu Tan. A probabilistic framework based on kde-gmm hybrid model (kghm) for moving object segmentation in dynamic scenes. In *In Workshop on Visual Surveillance, ECCV 2008*, 2008.

[56] R.M. Luque, E. Dominguez, E.J. Palomo, and J. Munoz. A neural network approach for video object segmentation in traffic surveillance. In Aurelio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, volume 5112 of *Lecture Notes in Computer Science*, pages 151–158. Springer Berlin Heidelberg, 2008.

[57] R.M. Luque, D. Lopez-Rodriguez, E. Dominguez, and E.J. Palomo. A dipolar competitive neural network for video segmentation. In Hector Geffner, Rui Prada, Isabel Machado Alexandre, and Nuno David, editors, *Advances in Artificial Intelligence - IBERAMIA 2008*, volume 5290 of *Lecture Notes in Computer Science*, pages 103–112. Springer Berlin Heidelberg, 2008.

[58] R.M. Luque, D. Lopez-Rodriguez, E. Merida-Casermeiro, and E.J. Palomo. Video object segmentation with multivalued neural networks. In *Hybrid Intelligent Systems, 2008. HIS '08. Eighth International Conference on*, pages 613–618, Sept 2008.

[59] D. Manjula M. Sivabalakrishnan. Adaptive background subtraction in dynamic environments using fuzzy logic. *International Journal of Image Processing*, 4(1), 2010.

[60] R. Ma, Liyuan Li, Weimin Huang, and Qi Tian. On pixel count based crowd density estimation for visual surveillance. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 170–173, 2004.

[61] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *Image Processing, IEEE Transactions on*, 17(7):1168–1177, July 2008.

[62] L. Maddalena and A. Petrosino. The sobs algorithm: What are the limits? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 21–26, June 2012.

[63] Lucia Maddalena and Alfredo Petrosino. Multivalued background/foreground separation for moving object detection. In Vito Di Gesu, SankarKumar Pal, and Alfredo Petrosino, editors, *Fuzzy Logic and Applications*, volume 5571 of *Lecture Notes in Computer Science*, pages 263–270. Springer Berlin Heidelberg, 2009.

[64] A.N. Marana, L.F. Costa, R.A. Lotufo, and S.A. Velastin. On the efficacy of texture analysis for crowd monitoring. In *Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI '98. International Symposium on*, pages 354–361, 1998.

[65] A.N. Marana, L. da Fontoura Costa, R.A. Lotufo, and S.A. Velastin. Estimating crowd density with minkowski fractal dimension. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3521–3524, 1999.

[66] A.N. Marana, S.A. Velastin, L.F. Costa, and R.A. Lotufo. Estimation of crowd density using image processing. In *Image Processing for Security Applications (Digest No.: 1997/074), IEE Colloquium on*, 1997.

[67] C. Marghes, T. Bouwmans, and R. Vasiu. Background modeling and foreground detection via a reconstructive and discriminative subspace learning approach. In *International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2012*, July 2012.

[68] G. Mateos and G.B. Giannakis. Sparsity control for robust principal component analysis. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pages 1925–1929, Nov 2010.

[69] Andrzej Materka and Michal Strzelecki. Texture analysis methods – a review. Technical report, INSTITUTE OF ELECTRONICS, TECHNICAL UNIVERSITY OF LODZ, 1998.

[70] N.J.B. McFarlane and C.P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3):187–193, 1995.

[71] S. Molina-Giraldo, J. Carvajal-Gonzalez, A.M. Alvarez-Meza, and G. Castellanos-Dominguez. Video segmentation framework based on multi-kernel representations and feature relevance analysis for object classification. In Ana Fred and Maria De Marsico, editors, *Pattern Recognition Applications and Methods*, volume 318 of *Advances in Intelligent Systems and Computing*, pages 273–283. Springer International Publishing, 2015.

[72] Dibyendu Mukherjee and Q.M. JonathanWu. Real-time video segmentation using student'stmixture model. *Procedia Computer Science*, 10(0):153 – 160, 2012. {ANT} 2012 and MobiWIS 2012.

[73] Esteban Jose Palomo, Enrique Dominguez, RafaelMarcos Luque, and Jose Munoz. Image hierarchical segmentation based on a ghsom. In ChiSing Leung, Minho Lee, and JonathanH. Chan, editors, *Neural Information Processing*, volume 5863 of *Lecture Notes in Computer Science*, pages 743–750. Springer Berlin Heidelberg, 2009.

[74] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, 2001.

[75] F. Porikli and C. Wren. Change detection by frequency decomposition: Waveback. In *International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2005*, 2005.

[76] V. Rabaud and S. Belongie. Counting crowded moving objects. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 705–711, 2006.

[77] H. Rahmalan, M.S. Nixon, and J.N. Carter. On crowd density estimation for surveillance. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 540–545, 2006.

[78] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, December 2010.

[79] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[80] CarlEdward Rasmussen. Gaussian processes in machine learning. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Ratsch, editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 63–71. Springer Berlin Heidelberg, 2004.

[81] J. Rosell-Ortega, G. Garcia-Andreu, A. Rodas-Jorda, and V. Atienza-Vanacloig. A combined self-configuring method for object tracking in colour video. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2081–2084, Aug 2010.

[82] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *Digital Image Computing: Techniques and Applications, 2009. DICTA '09.*, pages 81–88, 2009.

[83] David Ryan, Simon Denman, Sridha Sridharan, and Clinton Fookes. Scene invariant crowd counting and crowd occupancy analysis. In Caifeng Shan, Fatih Porikli, Tao Xiang, and Shaogang Gong, editors, *Video Analytics for Business Intelligence*, volume 409 of *Studies in Computational Intelligence*, pages 161–198. Springer Berlin Heidelberg, 2012.

[84] J. Rymel, J. Renno, D. Greenhill, J. Orwell, and G.A. Jones. Adaptive eigen-backgrounds for object detection. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 1847–1850 Vol. 3, Oct 2004.

[85] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1778–1792, Nov 2005.

[86] R. Sivalingam, A. De Souza, M. Bazakos, R. Miezianko, V. Morellas, and N. Papanikolopoulos. Dictionary learning for robust background modeling. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4234–4239, May 2011.

[87] Chris Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2, 1999.

[88] Alireza Tavakkoli, Mircea Nicolescu, and George Bebis. A novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Paolo Remagnino, Ara Nefian, Gopi Meenakshisundaram, Valerio Pascucci, Jiri Zara, Jose Molineros, Holger Theisel, and Tom Malzbender, editors, *Advances in Visual Computing*, volume 4291 of *Lecture Notes in Computer Science*, pages 40–49. Springer Berlin Heidelberg, 2006.

[89] H. Tezuka and T. Nishitani. A precise and stable foreground segmentation using fine-to-coarse approach in transform domain. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2732–2735, Oct 2008.

[90] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261 vol.1, 1999.

[91] Du-Ming Tsai and Shia-Chih Lai. Independent component analysis-based background subtraction for indoor surveillance. *Image Processing, IEEE Transactions on*, 18(1):158–167, Jan 2009.

[92] R. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1986)*, pages 364–374, 1986.

[93] Junxian Wang, G. Bebis, and R. Miller. Robust video-based surveillance by integrating target detection with tracking. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 137–137, June 2006.

[94] Naiyan Wang and Dit-Yan Yeung. Bayesian robust matrix factorization for image and video processing. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1785–1792, Dec 2013.

[95] Zhe Wang, Hong Liu, Yueliang Qian, and Tao Xu. Crowd density estimation based on local binary pattern co-occurrence matrix. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 372–377, 2012.

[96] Christopher R. Wren and Fatih Porikli. Waviz: Spectral similarity for object detection. In *In IEEE International Workshop on Performance Evaluation of Tracking & Surveillance*, 2005.

[97] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, Jul 1997.

[98] Mei Xiao, Chongzhao Han, and Xin Kang. A background reconstruction for dynamic scenes. In *Information Fusion, 2006 9th International Conference on*, pages 1–7, July 2006.

[99] Huan Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. *Information Theory, IEEE Transactions on*, 58(5):3047–3064, May 2012.

[100] Zhifei Xu, Irene Yu-Hua Gu, and Pengfei Shi. Recursive error-compensated dynamic eigenbackground learning and adaptive background subtraction in video. *Optical Engineering*, 47(5):057001–057001–11, 2008.

[101] Beibei Zhan, Dorothy N. Monekosso, Paolo Remagnino, Sergio A. Velastin, and Li-Qun Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, 2008.

[102] Hongxun Zhang and De Xu. Fusing color and texture features for background model. In Lipo Wang, Licheng Jiao, Guanming Shi, Xue Li, and Jing Liu, editors, *Fuzzy Systems and Knowledge Discovery*, volume 4223 of *Lecture Notes in Computer Science*, pages 887–893. Springer Berlin Heidelberg, 2006.

[103] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000.

[104] Cong Zhao, Xiaogang Wang, and Wai-Kuen Cham. Background subtraction via robust dictionary learning. *EURASIP Journal on Image and Video Processing*, 2011(1):972961, 2011.

[105] Tao Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, 2003.

[106] Xiaowei Zhou, Can Yang, and Weichuan Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):597–610, March 2013.

[107] Juhua Zhu, S.C. Schwartz, and B. Liu. A transform domain approach to real-time foreground segmentation in video sequences. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 2, pages 685–688, March 2005.

A

Clasificación de algoritmos BGS

En este apéndice se describe en detalle la clasificación de los segmentadores frente-fondo introducida en la sección 2.3.1.2 del Estado del Arte. Bouwmans [7] clasifica los métodos de extracción de fondo en dos grandes grupos:

1. Modelos Tradicionales: en este grupo se incluyen los primeros modelos utilizados en este campo y son, por lo general bastante básicos. Permiten lidiar con situaciones críticas específicas y suelen ser fáciles de implementar.

2. Modelos Recientes: a diferencia de los anteriores, estos modelos son más sofisticados y robustos para tratar escenarios complejos. La mayoría de ellos, necesita aún mejoras para poder ser incrementales y funcionar en tiempo real.

Cada uno de estos grupos está compuesto a su vez, por una gran cantidad de modelos que se pueden agrupar en otras clases. A continuación, se presenta de manera detallada dicha clasificación.

- **Modelos Tradicionales:** este grupo se puede dividir en varias categorías: modelos básicos, modelos estadísticos, modelos de agrupamiento, modelos de redes neuronales y modelos de estimación.

- **Modelos Básicos:** Son los primeros que se desarrollaron y presentan muy poca complejidad. Generan el modelo de fondo a partir de la media, la mediana [70], o análisis de histogramas en el tiempo, y clasifican los píxeles umbralizando la diferencia entre el fondo y la imagen actual.

- **Modelos Estadísticos:** Emplean funciones estadísticas para generar el modelo. Aportan robustez frente a cambios de iluminación y fondos dinámicos [6]. Estos métodos se pueden clasificar a su vez en modelos Gaussianos, modelos basados en modelos de soporte vectorial y modelos basados en aprendizaje subespacial. Los dos primeros, son los más apropiados para tratar con fondos dinámicos y el último funciona mejor en escenas con cambios de iluminación. Se puede decir, que los modelos estadísticos son los más utilizados ya que tienen un buen rendimiento con un coste computacional moderado.

  - Modelos Gaussianos: Con el objetivo de representar el fondo de la imagen, estos modelos asumen que las diferentes intensidades de un píxel a lo largo del tiempo se pueden modelar como una función Gaussiana. Muchos autores han empleado una única Gaussiana [44, 97], sin embargo, un modelo unimodal no puede funcionar correctamente con fondos dinámicos. Por este motivo, otros autores han utilizado mezclas de Gaussianas (**MOG**) [87]. También, se han desarrollado muchas otras mejoras para adaptar el modelo a situaciones críticas como las descritas en la sección anterior (ver 2.3.1.1) o para aumentar su eficiencia [24, 35, 85].

  - Modelos basados en modelos de soporte vectorial (sus siglas en inglés, **SVM**): Se trata de modelos estadísticos más sofisticados como **SVM** (*Support Vector Machine*) [53], **SVR** (*Support Vector Regression*) [93] o **SVDD** (*Support Vector Data Description*) [88].

  - Modelos basados en aprendizaje subespacial (*subspace learning models*): Los métodos de aprendizaje subespacial utilizan *Principal Component Analysis* (**SL-PCA**) y lo aplican sobre N imágenes para construir el modelo de fondo, que se representa mediante la imagen promedio y la matriz de proyección que comprende los primeros p autovectores significativos del **PCA**. La segmentación se realiza calculando la diferencia entre la

imagen de entrada y su reconstrucción. Numerosos autores han empleado este tipo de modelos y han realizado mejoras significativas sobre la idea inicial [36, 48, 84, 91, 100]. Se puede consultar el resumen realizado en [5] sobre este tipo de algoritmos.

- **Modelos de Agrupamiento:** Este tipo de modelos asume que cada píxel del *frame* se puede representar temporalmente por clústeres. Dentro de este grupo se hallan los algoritmos de *K-means* [9], modelos de *Codebooks* [45] o los métodos de *clustering* secuencial básicos [98]. Aparentemente, estos algoritmos se adaptan bien a fondos dinámicos y al ruido procedente de la compresión de vídeo.

  - Modelos *K-means*: Estos algoritmos asignan un grupo de clústeres a cada píxel del *frame*. La inicialización de realiza *off-line*. Los clústeres se ordenan de acuerdo a la probabilidad con la que modelan el fondo y se adaptan para hacer frente a variaciones del fondo y de iluminación. Los píxeles entrantes, se asignan a un clúster y se clasifican como fondo o no, según el clúster se considere parte del fondo o no.

  - Modelos de *Codebooks*: Para cada píxel, se crea un *codebook* que se compone de una o más *codewords* en base a una métrica de distorsión de color junto con límites de brillo. El número de *codewords* es distinto en función de los cambios sufridos por el píxel.

  - Modelos de *agrupamiento* secuencial básico: Estos métodos se basan en la idea de que todo aquello que sea parte del fondo debe aparecer en la escena durante un largo período de tiempo Las intensidades de los píxeles se clasifican según un modelo de clúster y posteriormente se calculan los centros y probabilidades de aparición de cada clúster. Finalmente, los clústeres con una probabilidad de aparición superior a un umbral se clasifican como fondo.

- **Modelos de Redes Neuronales (*Neural Networks* o NN):** En todos los modelos de este grupo, el fondo de la imagen se representa mediante medias de los pesos de un sistema de redes neuronales entrenado apropiadamente con N *frames* limpias. El sistema de redes neuronales aprende a clasificar los pí-

xeles como fondo o frente. Dentro de este grupo se pueden diferenciar varias categorías que utilizan sistemas de redes neuronales concretos como son: las redes neuronales de regresión general (**GNN**) [17], redes neuronales multi-evaluadas (**MNN**) [58], redes neuronales competitivas (**CNN**) [56], dipolar competitivas (**DCNN**) [57], mapas de autoorganizado (**SONN**) [61, 62], y mapas de autoorganizado de crecimiento jerárquico (**GHSONN**) [73]. Los algoritmos más destacados por sus resultados no sólo dentro de este grupo, sino también de todo el Estado del Arte utilizan **SONN** y se denominan *Self Organizing Background Subtraction* (**SOBS**) [61] y **SC-SOBS** (*Spatial Coherence* **SOBS**) [62].

- **Modelos de Estimación:** En los métodos de estimación el fondo de la imagen se determina utilizando un filtro. Cualquier píxel que se desvíe significativamente de su valor predicho se considera frente. Este filtro puede ser un filtro de Wiener [90], un filtro de Kalman [43] o un filtro de Chebychev [12]. En general estos algoritmos han demostrado funcionar bastante bien con cambios de iluminación.

- **Modelos Recientes:** Los modelos de extracción de fondo recientes se pueden dividir en las siguientes categorías: modelos de fondo estadísticos avanzados, modelos de fondo difusos, modelos de aprendizaje subespacial discriminatorio, modelos **RPCA**, modelos de minimización *low-rank*, modelos *sparse* y modelos de dominio transformado. A continuación, se explica cada uno de ellos.

  - **Modelos de Fondo Estadísticos Avanzados:** se trata de modelos estadísticos igual que los vistos anteriormente, pero más elaborados. Dentro de estos aparecen:

    - Modelos de mezclas: Se emplean distribuciones distintas a **GMM** (*Gaussian Mixture Model*) como pueden ser *Student-t Mixture Model* (**STMM**), [33, 72] o *Dirichelet Mixture Model* (**DMM**) [34]. En [72] se utiliza un algoritmo basado en **STMM** que presenta una gran robustez frente al ruido y permite modelar el fondo de manera bastante precisa con objetos del frente lentos y fondos dinámicos. También los algoritmos basados en **DMM** han demostrado ser muy robustos en entornos con fondos dinámi-

cos.

- ▷ Modelos híbridos: Estos modelos unen las ventajas de los modelos no paramétricos regionales (**KDE**) y **GMM** (*Gaussian Mixture Model*) para aproximar la distribución de color del fondo de la imagen. El color del frente se aprende de los píxeles vecinos de la imagen previa [20, 55].
- ▷ Modelos no paramétricos: Hay en este grupo 2 algoritmos, **ViBe** y **PBAS**.
  - ◇ **ViBe** (*Visual Background Extractor*): Algoritmo propuesto en [3] que construye el modelo de fondo mediante la agregación de valores observados para cada localización de píxel. Este algoritmo introduce varias innovaciones importantes en este campo. Una de ellas es la consistencia espacial que genera con los píxeles vecinos. También es notable el proceso de inicialización del modelo, ya que desde el segundo *frame* se dispondría de un modelo de fondo para realizar la segmentación. Sin embargo, el algoritmo presenta también algunas desventajas como pueden ser problemas para modelar el fondo en escenarios con fondos oscuros o cambiantes y sombras.
  - ◇ **PBAS** (*Pixel-Based Adaptive Segmenter*): Este algoritmo [37] genera el modelo de fondo a partir de las últimas observaciones de los píxeles. **PBAS** se compone de diversas partes, pero el elemento central es el bloque de decisión que determina si es frente o no basándose en la imagen actual y el fondo. La decisión se basa en un umbral a nivel de píxel. El fondo se actualiza a lo largo del tiempo con el objetivo de adaptarse a los cambios de acuerdo con un nuevo parámetro: la tasa de aprendizaje, también a nivel de píxel. Ambos umbrales, permiten obtener muy buenos resultados con fondos dinámicos, por lo que **PBAS** es uno de los mejores algoritmos del Estado del Arte.
- ▷ Modelos multi-kernel: Dentro de esta clase, se puede situar el algoritmo desarrollado por Molina-Giraldo [71] que utiliza múltiples kernels y representaciones de color para crear un espacio que represente mejor las características. Se emplea un algoritmo de agrupamiento sobre el espacio de características para clasificar los píxeles como frente o fondo.

- **Modelos Difusos (*Fuzzy models*):** Estos métodos introducen diferentes conceptos *fuzzy* o difusos en los pasos del proceso de extracción de fondo para solventar las imprecisiones o incertidumbres introducidas por situaciones complejas [4].

  - Modelado de fondo difuso (*Fuzzy Background Modeling*): Estos métodos se utilizan principalmente para hacer frente a fondos multimodales. Por lo general, se emplea un algoritmo **GMM** [87], sin embargo, la secuencia de entrenamiento puede contener insuficientes datos o ruido. Para solucionar este problema, se han implementado otros algoritmos como *Type-2 Fuzzy Mixture of Gaussians* (**T2F-MOG**) [22, 23]. El Baf propuso dos algoritmos: **T2-FMOG-UM** y **T2-FMOG-UV,** que se utilizan para la incertidumbre sobre la media y sobre la varianza respectivamente, siendo **T2-FMOG-UM** el más robusto. Otra alternativa aparece en [46], donde se implementa un algoritmo denominado *Fuzzy C-Means Clustering Model,* que utiliza como característica un histograma de colores difusos que permite atenuar las variaciones de color debidas al movimiento del fondo.

  - Detección de fondo difusa (*Fuzzy Foreground Detection*): En este grupo, aparecen algoritmos que utilizan una función lineal de saturación para facilitar la clasificación de los píxeles como frente o fondo [81]. Otros, agregan características de color o textura, que se utilizan para calcular medidas de similitud que posteriormente se integran con la integral de Sugeno [102] o con la integral de Choquet [21] y, finalmente, se detectan los objetos en movimiento umbralizando los resultados. Los métodos de este grupo son bastante robustos a cambios de iluminación y sombras.

  - Actualización del fondo difusa (*Fuzzy Background Maintenance*): La idea principal de estos algoritmos es actualizar el fondo de la imagen teniendo en cuanta la pertenencia de los píxeles al frente o al fondo. Esto se puede hacer de 2 maneras: adaptando la tasa de aprendizaje siguiendo la clasificación de los píxeles con un método difuso [63] o combinando de manera difusa dos reglas de actualización distintas [2]. Estos métodos permiten obtener mejores resultados con cambios de iluminación y sombras.

- ▷ Características difusas (*Fuzzy Features*): Dentro de este grupo hay en la literatura 3 aproximaciones distintas de los mismos autores y cada una de ellas emplea características difusas diferentes [13, 14, 15].
- ▷ Post-procesado difuso: Este método se puede aplicar directamente sobre los resultados de otros algoritmos. Para mejorar la detección, se puede hacer una inferencia difusa entre la máscara del frente de la imagen anterior y actual [59].

En general, los modelos difusos funcionan bastante bien con cambios de iluminación y fondos dinámicos.

- ○ **Modelos de Aprendizaje Subespacial Discriminatorios y Mixtos:** Los modelos discriminatorios de aprendizaje subespacial permiten la inicialización y clasificación de los píxeles de manera robusta y supervisada, mientras que los métodos de aprendizaje subespacial reconstructivos no requieren supervisión, aunque funcionan peor.
  - ▷ Modelo subespacial discriminatorio: Se puede destacar el algoritmo propuesto en [27], denominado **IMMC** (*Incremental Maximum Margin Criterion*) que funciona mejor que los métodos reconstructivos, pero a cambio requiere de imágenes de *ground truth* para la inicialización del fondo.
  - ▷ Modelo subespacial mixto: Hay un algoritmo en la literatura [67] que combina un método reconstructivo (**PCA**) con otro discriminatorio (**LDA**) para modelar el fondo de manera más robusta que los métodos reconstructivos y discriminatorios por separado.
- ○ **Modelos Subespaciales Robustos (*RPCA*):** Estos modelos se basan en minimización *low-rank* y *sparse descomposition*. En este grupo se incluyen los algoritmos *Robust Principal Component Analysis* (**RPCA**) entre los que se encuentran ***RPCA*** *via Principal Componens Pursuit [10],* ***RPCA*** *via Outlier Pursuit* [99], ***RPCA*** *via Sparsity Control* [68], etc. En general, esta familia de algoritmos no ha demostrado funcionar bien con todas las dificultades que suponen las escenas reales.
- ○ **Modelos de Minimización *low-rank* (LRM):** Se trata de algoritmos muy

útiles en tareas de minería de datos y que han sido recientemente mejorados para ganar robustez frente a valores atípicos (*outliers*), que tradicionalmente han afectado a su funcionamiento [94, 106].

- **Modelos *Sparse*:** Se utiliza una *sparse aproximation* para generar un vector *sparse* multidimensional a partir de datos observados de una dimensión mucho mayor. Estos algoritmos se pueden dividir en varias categorías: *compressive sensing models* [51], *structured sparsity models* [39], *dynamic group sparsity models* [18, 86, 104] y *dictionary models.*

- **Modelos de Dominio Transformado:** Estos métodos buscan separar el fondo y el frente de la imagen en un dominio diferente. Para ello utilizan distintas transformadas.

  - Transformada rápida de Fourier (**FFT**): En [96] se estima el modelo de fondo mediante la captura de marcas espectrales de fondos multimodales utilizando **FFT**. Los cambios en la escena se detectan por la inconsistencia con las marcas espectrales. Los resultados muestran robustez frente a objetos de bajo contraste y escenas dinámicas.

  - Transformada discreta del coseno (**DCT**): En [75] se presenta un algoritmo que representa el fondo mediante la descomposición frecuencial del historial de los píxeles de la imagen. Se elabora un mapa de distancias a partir de los coeficientes **DCT** del fondo y de las imágenes actuales y posteriormente, se aplica un umbral sobre el mapa para detectar el frente. Este algoritmo tiene buenos resultados con fondos dinámicos. Existen otros ejemplos similares en el Estado del Arte [107] que han demostrado ofrecer resultados igual de precisos y con menor coste computacional que los de dominio espacial.

  - Transformada de Walsh (**WT**): Tezuka y Nishitani [89] proponen modelar el fondo aplicando **MOG** sobre varios tamaños de bloques utilizando **WT**. Los resultados demuestran que la naturaleza espectral de WT reduce el coste computacional. Además, los autores desarrollaron una transformada rápida y selectiva de Walsh (**SFWT**).

- ▷ Transformada Wavelet (**WT**): Hay en el Estado del Arte varios algoritmos que han empleado **WT.** Por ejemplo, Gao en [30] propuso un algoritmo para modelar el fondo basado en Marr wavelet kernel y que emplea una característica basada en transformadas wavelet binarias discretas, obteniendo resultados mejores que **MOG** en escenas reales. Guan en [32] introdujo el uso de *Dyadic Wavelet* para detectar el frente, que descompone en componentes wavelet multiescala la diferencia entre el fondo y las imágenes actuales. También en [29, 41] se utiliza la transformada wavelet para la extracción de fondo.

- ▷ Transformada de Hadamard (**HT**): En [1] se propone un algoritmo que permite inicializar el fondo rápidamente, ya que compara las marcas espectrales obtenidas con **HT** y extraídas de los bloques de los primeros *frames.*

Como se ha visto hasta ahora, hay multitud de algoritmos en el Estado del Arte que pretenden solventar la segmentación mediante modelado de fondo. Sin embargo, prácticamente ninguno de ellos es capaz de afrontar con éxito toda la problemática expuesta en la sección 2.3.1.1 que aparece en escenarios reales. Según el tipo de escena y las condiciones en que se haya capturado la secuencia podrían aparecer diversos factores tales como: ruido, cambios de iluminación, sombras, fondos dinámicos, etc., que afectan de manera significativa los resultados de la segmentación. Por este motivo, estos métodos se encuentran en continua evolución y abordan la problemática del modelado de fondo mediante la aplicación de diversos y complejos modelos matemáticos que buscan, no sólo una aproximación más precisa, sino también más eficiente y con un coste computacional menor.

De todos los algoritmos presentados, **ViBe** [3], **PBAS** [37], **SOBS** [61] y **SC-SOBS** [62] son los mejor valorados por su capacidad para lidiar con todos los factores propios de escenarios reales.

B

Glosario de acrónimos

- **BGS**: Background Substraction
- **DBP**: Dependent Body Parts
- **DCT**: Discrete Cosine Transform
- **DEBP**: Dependent Extended Body Parts
- **DEBP-P**: Dependent Extended Body Parts Post-Processed
- **DMM**: Dirichelet Mixture Model
- **FFT**: Fast Fourier Transform
- **GMM**: Gaussian Mixture Model
- **GP**: Gaussian Process
- **GPML**: Gaussian Processes for Machine Learning
- **HOG**: Histogram of Oriented Gradients
- **HT**: Hadamard Transform
- **IBP**: Independent Body Parts

- **IEBP**: Independent Extended Body Parts
- **IMMC**: Incremental Maximum Margin Criterion
- **KDE**: Kernel Density Estimation
- **MAE**: Mean Absolute Error
- **MOG**: Mixture of Gaussians
- **MSE**: Mean Squared Error
- **NN**: Neural Network
- **PBAS**: Pixel-Based Adaptative Segmenter
- **PCA**: Principal Component Analysis
- **RPCA**: Robust Principal Component Analysis
- **SFWT**: Fast Fourier Transform
- **SOBS**: Self Organizing Background Subtraction
- **STMM**: Student-t Mixture Model
- **SVDD**: Support Vector Data Description
- **SVM**: Support Vector Machine
- **SVR**: Support Vector Regression
- **ViBe**: Visual Background Extractor
- **WT**: Walsh Transform
- **WT**: Wavelet Transform

C

Presupuesto

1) **Ejecución Material**

- Compra de ordenador personal (Software incluido) — 2.000,00 €
- Alquiler de impresora láser durante 6 meses — 260,00 €
- Material de oficina — 150,00 €
- Total de ejecución material — 2.400,00 €

2) **Gastos generales**

- sobre Ejecución Material — 352,00 €

3) **Beneficio Industrial**

- sobre Ejecución Material — 132,00 €

4) **Honorarios Proyecto**

- 1800 horas a 15 €/ hora — 27000,00 €

5) Material fungible

▪ Gastos de impresión	280,00 €
▪ Encuadernación	200,00 €

6) Subtotal del presupuesto

▪ Subtotal Presupuesto	32.774,00 €

7) I.V.A. aplicable

▪ 21 % Subtotal Presupuesto	6.882,54 €

8) Total presupuesto

▪ Total Presupuesto	39.656,54 €

Madrid, Mayo 2015

El Ingeniero Jefe de Proyecto

Fdo.: Rosely Sánchez Ricardo

Ingeniero de Telecomunicación

D

Pliego de condiciones

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un *Estimación de la densidad de personas basado en segmentación frente-fondo y segmentación fondo-persona*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales.

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus

facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares.

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.